VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

EFEKTIVNÍ ZABEZPEČENÍ KOMUNIKACE BEZDRÁTOVÝCH NÍZKO-ENERGETICKÝCH MODERNÍCH TECHNOLOGIÍ

EFFICIENT SECURITY OF MODERN LOW-POWER WIRELESS COMMUNICATION TECHNOLOGIES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Vojtěch Blažek**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Radek Fujdiak, Ph.D.**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Vojtěch Blažek ***ID:*** 186490
Ročník: 3 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Efektivní zabezpečení komunikace bezdrátových nízko-energetických moderních technologií

POKYNY PRO VYPRACOVÁNÍ:

Hlavním úkolem studenta bude implementovat energeticky vhodnou a bezpečnou šifru určenou pro velmi omezené podmínky z pohledu zdrojů – paměti, energie, výkonu, doby výpočtu, aj. Vhodnost šifry vyplyne z rozsáhlého měření efektivity zprovozněných šifer na platformě Arduino. Bude se jednat převážně o šifry z řad základních (např. OTP) a symetrických (proudových) šifer. Bude provedena nejen analýza slabin nezabezpečeného systému, ale také energetická analýza jednotlivých šifer a tedy i následné navýšení režie v rámci přenosu dat. Výstupem tak bude bezpečné a nízkoenergetické komunikační řešení.

DOPORUČENÁ LITERATURA:

[1] Amel Technology. Smart Everything FOX: User Guide (Ver. 1.0), 2015.

[2] SigFox. SigFox: Technical Overview. May, 2017.

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 29.5.2018

Vedoucí práce: Ing. Radek Fujdiak, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato práce se zabývá problematikou zabezpečení komunikace v Sigfox sítích. To zahrnuje prostudování parametrů a možností sítě a následně analýzu vhodných kryptosystémů. Práce obsahuje popis kryptografie jako takové a popis kryptosystémů, které by bylo možné implementovat do koncových zařízení. Z vybraných kryptosystémů jsou na základě požadovaných vlastností vybrány tři, které jsou uvedeny do praxe za pomoci vývojového kitu, založeném na platformě Arduino a následně jsou provedena experimentální měření, kvůli zjištění základních vlastností zařízení.

KLÍČOVÁ SLOVA

Internet Věcí, Kryptografie, LPWAN, Sigfox, Šifrování, One-Time Pad

ABSTRACT

This thesis deals with communication security in Sigfox networks. That includes studying the network's parameters and capabilities and then analysing the appropriate cryptosystems. The thesis contains a description of cryptography as such and a description of cryptosystems that could be implemented in end devices. From the selected cryptosystems, three of the most appropriate are selected based on the required properties, which are put into practice on the Arduino-based development kit and then experimental measurements are made to determine the basic features of the device.

KEYWORDS

Internet of Things, Cryptography, LPWAN, Sigfox, Encryption, One-Time Pad

BLAŽEK, Vojtěch. *Efektivní zabezpečení komunikace bezdrátových nízko-energetických moderních technologií*. Brno, 2018, 46 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Radek Fujdiak, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Efektivní zabezpečení komunikace bezdrátových nízko-energetických moderních technologií“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Radkovi Fujdiakovi, Ph.D. za výborné odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Zabezpečování dat a chránění jejich obsahu před zneužitím je v dnešní době potřeba ve všech typech komunikací. Nezáleží jestli posíláte běžnou zprávu kolegovi nebo se zrovna přihlašujete do internetového bankovnictví. Všechna data by měla být do jisté míry chráněna a šifrována, protože nikdy nevíte kdo a jak je chce zneužít.

Tato práce je věnována návrhu zabezpečení komunikace bezdrátových nízkoenergetických moderních technologií. Především se tedy zabývá výběrem vhodných kryptografických algoritmů a implementací třech nejvhodnějších šifer do koncového zařízení, na kterém následně budou provedena základní měření pro zjištění vlastností zařízení.

Tato práce je dále rozdělena na šest kapitol. V kapitole první jsou ve zkratce popsány LPWAN technologie doplněné o tabulku srovnání nejznámějších technologií typu LPWAN. V druhé kapitole je popsána komunikační síť Sigfox. Přesněji je popsán popis modulace a komunikace Sigfoxu celkově a na závěr již nasazená zabezpečení při používání Sigfox sítě. Při popisu Sigfox sítě je klíčové ujasnit čtenáři, jaké má tato síť přenosové možnosti a omezení, což je později důležité při výběru šifry pro zabezpečení komunikace mezi koncovými zařízeními a uživatelským serverem. Třetí kapitola obsahuje popis kryptografie obecně a podkapitolu obsahující výběr šifer. Jsou v ní tedy zmíněny základní šifrovací postupy, je vysvětleno podle čeho se šifry rozdělují a v podkapitole se nachází stručný popis všech šifer, které jsme uvažovali ve výběru. Nejdůležitější částí třetí kapitoly je shrnutí, ve kterém je vysvětleno proč jsou vybrané finální šifry nejvhodnější pro tento typ komunikační sítě. Na tuto kapitolu navazuje kapitola *4 Popis vybraných šifer*. Ta obsahuje podrobný popis fungování, možností a omezení všech třech vybraných šifer. Tyto šifry jsou poté implementovány do vývojového kitu, což popisuje kapitola *5 Realizace zabezpečení*. Tato kapitola obsahuje popis použitého hardwaru a především popis funkcí jednotlivých implementací pro všechny tři šifry. Poslední kapitola práce se věnuje provedeným měřením. Ty mají za cíl zjistit energetickou náročnost provedených implementací šifer a rozhodnout která šifra je nejúspornější. Na základě výsledků těchto měření pak bude zvolena šifra, která by případně měla být nasazena do praxe.

1 ZÁKLADNÍ PŘEHLED TECHNOLOGIÍ TYPU LPWAN

LPWAN jsou radiové technologie, které se od jiných radiových technologií liší několika specifickými charakteristikami. Mezi hlavní požadavky u LPWAN technologií patří velký dosah, dlouhá životnost baterie a co nejnižší náklady na koncové zařízení. Obecně by se měl dosah v otevřené krajině pohybovat kolem 20 km, životnost baterie by měla být větší než 10 let, cena koncového zařízení by měla být nižší než 500 Kč (20 USD) a měli by být zajištěny nízké přenosové rychlosti kolem 200 b/s. Fyzická vrstva těchto technologií je řešená dvěma způsoby. Buď kmitočtové dělení do velmi úzkých kanálů (Ultra Narrow Band, UNB), kde jsou typickými představiteli systémy Sigfox a Telensa, nebo rozprostřené spektrum (Spread Spectrum, SS), které reprezentují zejména systémy LoRaWAN a RPMA [1].

V LPWAN technologiích se používá hvězdicová topologie, kde jedna základnová stanice pokrývá velkou plochu a obsluhuje velké množství koncových stanic. V praxi koncová zařízení vysílají naměřené hodnoty, výstrahy, potvrzení a back-end servery odesílají povely, potvrzení a aktualizace. Požadavky jednotlivých aplikací na přenosovou kapacitu, latenci, energetický rozpočet, četnost odesílání zpráv, zabezpečení, spolehlivost, možnosti aktualizace se budou lišit podle konkrétních požadavků na komerčně poskytnuté služby. Typické oblasti využití jsou tzv. Smart-city, infrastruktura, zdravotnictví, logistika, zemědělství, vzdálené monitorování, průmyslová výroba, zabezpečení a jiné [2]. Tabulka 1.1 uvádí nejrůznější technologie typu LPWAN spolu s jejich základními přenosovými vlastnostmi. Práce se však bude dále zabývat jen technologií Sigfox.

Tab. 1.1: Porovnání nejrozšířenějších LPWAN technologií (převzato z [3]).

Technologie	**Kmitočtové pásmo**	**Šířka kanálu**	**Přenosová rychlost**	**Maximální dosah**	**Standard**
HaLow	pod 1 GHz	1/2/4/8/16 MHz	0,15–18 Mb/s	2 km	IEEE 802.11ah
LoRa	pod 1 GHz	125/250 kHz	0,25–50 kb/s	5–15 km	LoRa Alliance
LTE-M	800/900 MHz	1,4 MHz	0,2–1 Mb/s	5–11 km	3GPP
Nwave	pod 1 GHz	200 Hz	100 b/s	7–10 km	proprietární
RPMA	2,4 GHz	1 MHz	156/624 kb/s	5–8 km	proprietární
Sigfox	pod 1 GHz	100 Hz	100 b/s	10–50 km	proprietární
Telensa PLANet	pod 1 GHz	250 Hz	60/500 b/s	2–8 km	proprietární
WAVIoT NB-Fi	pod 1 GHz	100 Hz	10–100 b/s	10–50 km	proprietární
Weightless-P	pod 1 GHz	12,5 kHz	0,2–100 kb/s	2–5 km	Weightless SIG

2 SIGFOX

Sigfox je komunikační bezdrátový systém pro energeticky nenáročný přenos malého množství dat na vzdálenosti až několika kilometrů. Typickými oblastmi aplikací této sítě v Evropě jsou parkovací senzory, Industry 4.0, SmartCity, zabezpečovací zařízení, logistika, sledování teplot při transportu a uskladnění, péče o seniory, měření srážek a průtoků na záplavových tocích a odečty vody, elektřiny, plynu apod. Sigfox byl vyvinutý hlavně jako spolehlivá přenosová síť, čehož se dosahuje opakovaným posíláním zpráv po sobě. Poskytuje také menší datovou přenosovou kapacitu v porovnání s technologií LoRa, což hlavně slouží k úspoře energie [5].

2.1 Popis modulace Sigfox

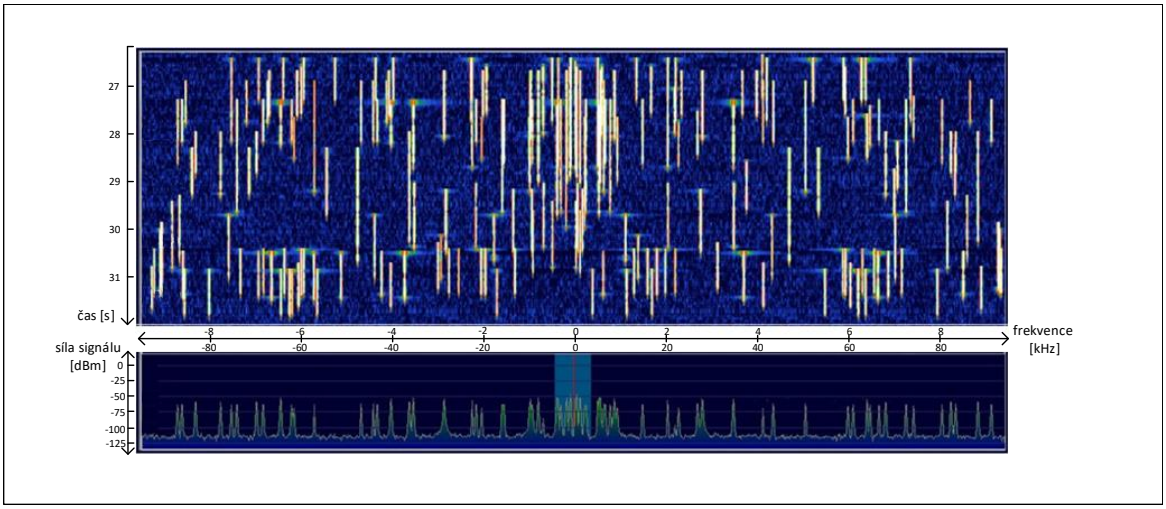
Sigfox pracuje v bezlicenčním ISM pásmu 868 MHz (906 MHz v USA). Protože v tomto pásmu nevysílají zařízení používající bluetooth, WiFi ani energomonitor, není těmito bezdrátovými technologiemi ovlivňována (rušena). Využívá se tzv. UNB pásmo pro vysílání jen krátkého pulsu dat s vysílacím výkonem omezeným na 100 mW a modulací pracující v 200 kHz veřejném pásmu. Každá přenášená zpráva v době přenosu zabírá šířku pásma 100 Hz a je přenášena rychlostí 100 nebo 600 bitů/s (v závislosti na regionu). Vysílaná zpráva využívá modulaci DBPSK, které stačí pro rychlost přenosu 1 bit/s jen frekvenční pásmo 1 Hz. Tedy když je šířka frekvenčního pásma zprávy 100 Hz, tak přenosová rychlost bude 100 bit/s [5].

Základní výhody DBPSK modulace [5]:

- snadné provedení (nasazení),
- nízká cena potřebných součástek,
- velká citlivost přijímacích stanic (možnost demodulace signálu blízkého úrovni šumu):
  - Při 100 bit/s je citlivost = –142 dBm.
  - Při 600 bit/s je citlivost = –134 dBm.
  - Teoretický spojovací součet dosahuje 163,3 dB.

Na Obr. 2.1 můžeme vidět přenos 210 UNB Sigfox signálů, které zabírají jen 4 % celkové kapacity a tento přenos proběhl bez kolizí [5]. Na vodorovné ose x, je pro dolní část obrázku vyjádřena frekvence v kilohertzích [kHz] a pro horní část obrázku frekvence ve stovkách hertzů [10^2 Hz]. V horní části je na svislé ose y vyjádřen čas, takže horní část obrázku ukazuje na jaké frekvenci a jak dlouho byly dané zprávy vysílány. Barva vyjadřuje vysílaný výkon (modrá minimální, bílá maximální povolený výkon 100 mW). Dolní část ukazuje celé využívané pásmo 200 kHz,

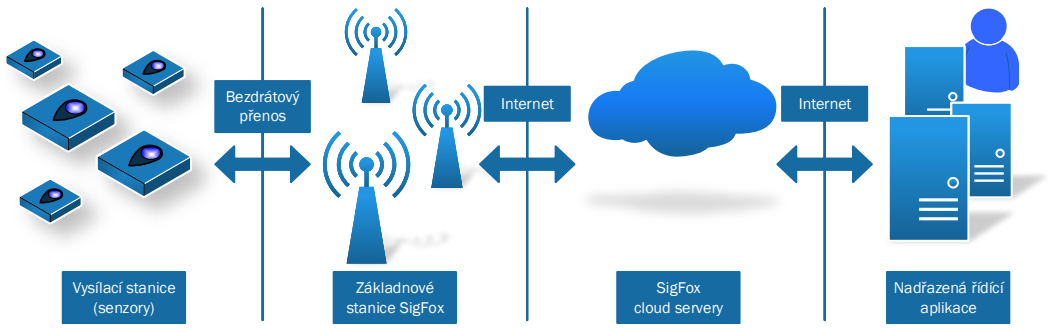
kde svislá osa ukazuje sílu signálu v dB. Z obrázku je také patrné, že odeslání jedné zprávy trvá přibližně jednu až dvě vteřiny.



Obr. 2.1: Sigfox spektrum (převzato z [6].)

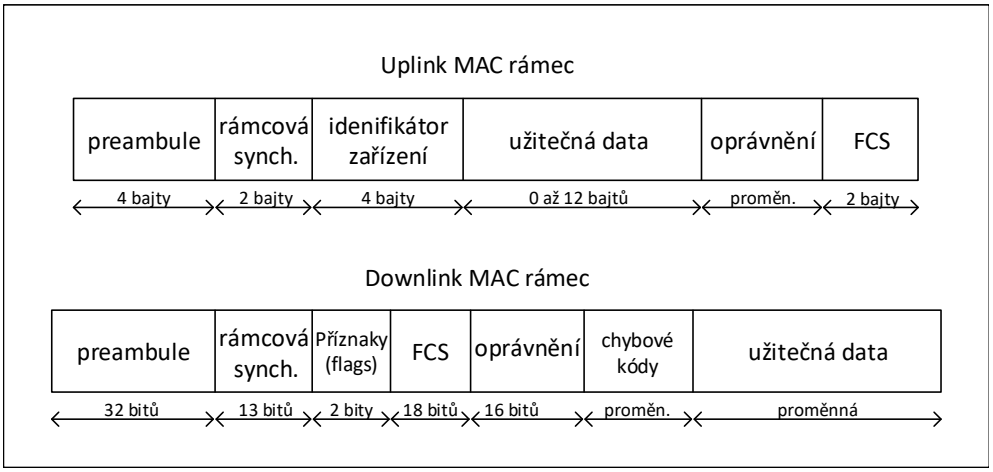
2.2 Komunikace v Sigfox

Samotná síť pro technologii Sigfox (Obr. 2.2) je založena na topologii hvězda a je budována na buňkovém principu: má své základnové stanice a buňky, které pokrývají určité oblasti. Každá jednotlivá stanice využívá místní Sigfox operátory, kteří následně obdržené zprávy posílají do Sigfox cloudu již přes TCP/IP internetovou komunikaci. Ten následně zprávy přetřídí a pošle obvykle po internetu do zákaznických zařízení a IT platforem.



Obr. 2.2: Architektura sítě Sigfox (převzato z [5]).

Pokud jde o adresování, tak jednotlivá koncová zařízení jsou identifikována interními identifikátory (Obr. 2.3), která jsou ekvivalentem sériových čísel. Lze to tedy připodobnit k MAC adrese každého ethernetového zařízení.



Obr. 2.3: Struktura rámce v Sigfox komunikaci (převzato z [5]).

Sigfox je omezen denně na 140 vysílacích a 4 zpětné zprávy (lze odeslat zprávu každých 10 m). Zatímco na sběr dat ze stacionárních objektů, jako jsou automaty a jiné stroje, to může plně stačit, ale například na funkci energomonitoru, kde se odesílají data přibližně každou minutu, to již nestačí.

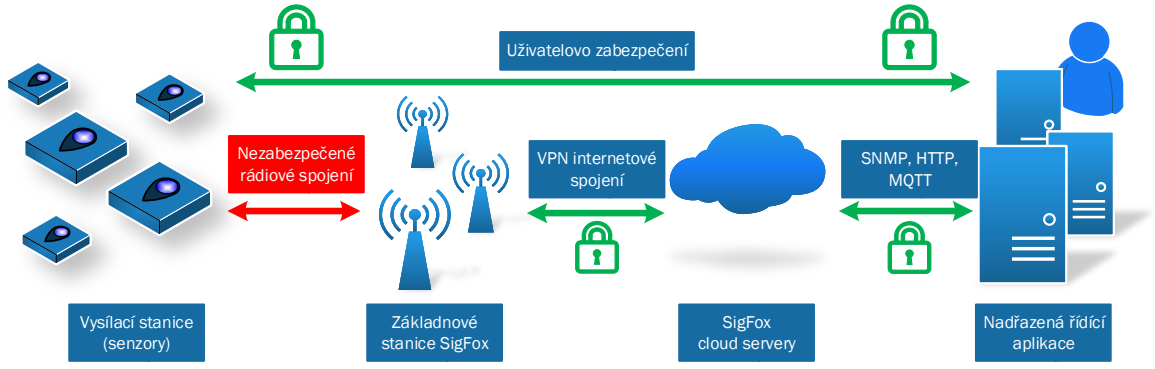
Celkový přenosový rámec tvoří pouze 26 bajtů, kde volitelně 0 až 12 bajtů je určeno pro užitečná data (payload). Tato zpráva může v praxi například reprezentovat 2 GPS souřadnice s přesností na 3 m, 6 hodnot naměřené teploty senzorů s rozsahem -100 °/+200 ° a s přesností 0.004 °, 12 měření radaru rychlosti s rozsahem až do 255 km/h, záznam až 96 dvoustavových signálů (např. typu den/noc, horký/studený, zapnuto/vypnuto, nízký/nedostatečný stav energie atd.)

Zpětný kanál, který již poslední verze Sigfoxu umožňuje, je pak datově omezený na 0 až 8 bajtů užitečných dat s časovým omezením na 4 zprávy denně. To by v praxi mělo dostačovat pro příkaz změny rozsahu senzoru, změny frekvence zasílání zpráv, zapnutí/vypnutí některé funkce apod. Tímto způsobem lze například na dálku aktivovat dočasně vypnutý, výkonnější způsob komunikace (např. GSM nebo WiFi).

Například pro srovnání jen záhlaví komunikačního rámce IP stacku má 40 bajtů. Zde je vidět, že pro přenos velmi malého množství dat je TCP/IP protokol velmi neekonomický, protože čím více dat se vysílá, tím více energie je pro přenos potřeba [5].

2.3 Zabezpečení Sigfox komunikace

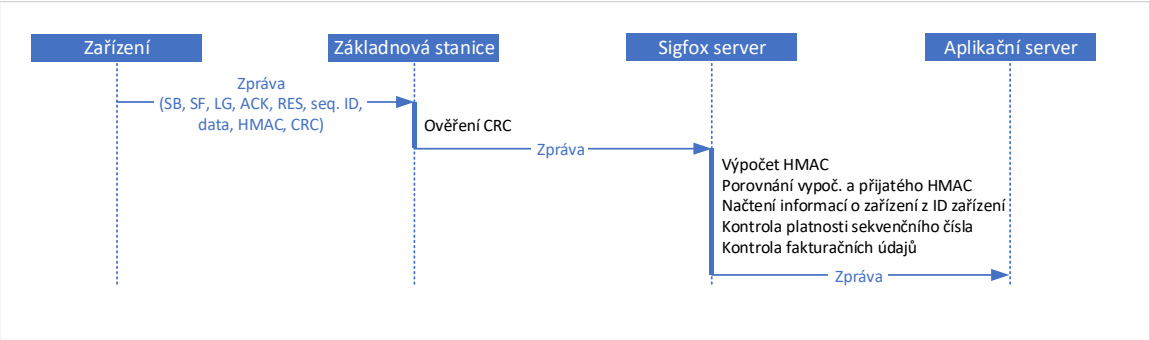
Sigfox používá určité mechanismy a nástroje k zabezpečení posílaných dat po síti (např. privátní klíče zařízení, ochrana proti replay útokům, message scrambling, sequencing a SSL certifikáty základnových stanic), ale neposkytuje bezpečnou komunikaci mezi koncovým zařízením a základnovými stanicemi (znázorněno na Obr. 2.4). Uživatelé by tedy měli vždy provádět dodatečné šifrování v rámci payloadu 12 bajtů [4][7]. Data jsou dále zranitelná na serverech Sigfox poskytovatele, kde k nim do jisté míry mají přístup někteří zaměstnanci Sigfox poskytovatele. Teprve až s využitím vlastního šifrování dat dostáváme opravdu bezpečnou end-to-end komunikaci.



Obr. 2.4: Znázornění zabezpečení Sigfox komunikace (převzato z [5]).

Jedním ze základních opatření proti útokům je sekvenční číslování zpráv. To spočívá jen v počítadle, které se s každou odeslanou zprávou inkrementuje. Vytváří ho vysílací stanice (koncové zařízení) a je ověřováno Sigfox podpůrným systémem. Slouží jako zabezpečení proti tzv. discard replay útokům, které by pak měli za následek zahazování zpráv z původního zařízení a přijímání zpráv podvržených [8].

Dále je zajištěna také integrita a autentizace posílaných zpráv pomocí ověřování MAC (Message Authentication Code). Každé koncové zařízení obsahuje již z výroby symetrický autentizační klíč (jinak také nazývaný Network Authentication Key - NAK). Každá zpráva, která má být odeslána nebo přijata, obsahuje kryptografickou známku vypočítanou právě na základě tohoto klíče a příslušné zprávy. Ověření těchto známek zajistí integritu zpráv a autentizaci odesílatele. Mezi Sigfox servery a nadřazenými aplikačními servery uživatelů se s zajištěním integrity a autentizace již spoléhá na Internetové protokoly jako jsou VPN a HTTPS [8].



Obr. 2.5: Zabezpečení Sigfox zprávy během komunikace (převzato z [8]).

Co se šifrování týče, tak Sigfox dává svým zákazníkům na výběr, buď si vytvořit vlastní end-to-end šifrování, a nebo se spolehnout na šifrovací řešení přímo od Sigfoxu využívající šifru AES-128 v CTR režimu. To však nevytváří úplné end-to-end šifrování, jelikož se zprávy dešifrují již na straně Sigfox serverů a poté se již nešifrované posílají skrze Internetové protokoly k aplikačním serverům uživatele. Pořád tedy přetrvává hrozba ze stran zaměstnanců Sigfox poskytovatele, která byla zmíněna výše. Toto řešení ze strany Sigfoxu je samozřejmě zpoplatněno. Cena služby je závislá od tříd zabezpečení (střední, vysoká, velmi vysoká), které se liší v způsobu zabezpečení citlivých informací v zařízení. Citlivá data potřebná k fungování všech výše zmíněných zabezpečení jsou následující:

- symetrický autentizační klíč NAK - potřebné k výpočtu MAC,
- ID zařízení - potřebné k identifikaci zařízení na Sigfox serveru,
- sekvenční číslo - musí být přítomné ve zprávách pro jejich přijetí serverem,
- šifrovací klíč - umožňuje zašifrování uživatelských dat (v případě využití šifrovací služby),
- počítadlo CTR - potřebné k šifrování v režimu CTR (v případě využití šifrovací služby).

3 KRYPTOGRAFIE

Kryptografie je technika používající se pro bezpečnou komunikaci mezi dvěma stranami za předpokladu přítomnosti strany třetí, se kterou nechceme sdílet obsah komunikace. Jedná se tedy o konstrukci a analýzu protokolů, které zamezují třetím stranám zjistit obsah zprávy a zároveň někdy zajišťují detekci změny dat během přenosu. Tyto protokoly musí zajistit bezpečnost posílaných informací a měli by splňovat aspekty jako jsou důvěrnost, autenticita, integrita dat a nepopiratelnost.

- Důvěrnost:
  - Cílem důvěrnosti je zajistit, že informace jsou přístupné pouze těm, kteří jsou k tomu oprávněni.
- Autenticita:
  - Cílem je zajištění schopnosti prokázat odesílatelovu identitu a zároveň aby se potencionální útočník nebyl schopen vydávat za někoho jiného.
- Integrita:
  - Cílem je zajistit data tak, aby s nimi nemohl útočník nijak manipulovat či je pozměnit na cestě od autora k příjemci bez vědomí příjemce. Útočník by neměl být schopen nahradit zprávu falešnou za legitimní.
- Nepopiratelnost:
  - Odesílatel by neměl být schopen popírat, že je opravdu autorem dané zprávy a že on ji skutečně odeslal.

Pokud odesílatel chce příjemci bezpečně poslat zprávu, může její obsah zašifrovat, a tím skrýt její skutečný obsah a transformovat ji do tzv. šifrového textu. Zpráva se tak stává nečitelnou pro všechny s výjimkou příjemce. Ten ji pak může následně dešifrovat, aby obnovil původní znění zprávy. Je běžné, že algoritmy pro šifrování a dešifrování jsou veřejně známé a může je kdokoliv používat, pouze šifrovací/dešifrovací klíče, které autor vytvořil a zašifroval s nimi příslušnou zprávu, jsou udržovány v tajnosti [9].

3.1 Základní šifrovací algoritmy

Mezi základní šifrovací algoritmy patří [10]:

- Jednoduché aritmetické operace:
  - Dvojici vstupních bitů či bloků bitů se pak přiřazuje výstupní bit či blok bitů na základě určitých aritmetických operací. Nejčastěji se používá operace součtu modulo dvě. Příklad šifry, která používá operace součtu modulo dvě, je Vernamova dokonalá šifra.
- Substituce:

  - Každý znak původní zprávy je nahrazen definovaným způsobem znakem jiným. Takto se dají nahrazovat i celé skupiny znaků.
- Transpozice:
  - Při transpozici se definovaným způsobem mění pořadí znaků ve zprávě.

3.2 Základní dělení kryptografických systémů

Kryptografické systémy lze rozdělit dle použití klíčů šifrovacího K_E a dešifrovacího K_D a to, zda jsou shodné či různé. Pokud jsou klíče různé, tak se v zásadě dělí na veřejný a utajený.

Klasické šifry

Klasické šifry jsou zvláštní třída šifer, které byly vynalezeny před mnoha lety a jsou specifické svojí funkčností. Výrazně se liší od moderních šifer a jejich bezpečnost je založena na jiných principech než jaké využívají dnešní šifry. Většina z nich se v moderní době již nepoužívá. Patří mezi ně např. Caesarova, Mřížková, Hillova či Vernamova šifra.

3.2.1 Symetrické kryptografické systémy

O symetrických kryptografických systémech mluvíme tehdy, když odesílatel i příjemce používají pro šifrování a dešifrování zprávy stejný klíč. Symetrické algoritmy samy o sobě jsou schopny zajistit důvěrnost zprávy (to znamená, že obsah zprávy je schopen dešifrovat jen sám autor a příjemce), ale k dosažení autenticity, integrity nebo nepopiratelnosti je potřeba použít odlišné techniky [9].

U symetrických systémů je kritické zaručení bezpečného přenosu klíče a poté jeho ochrana před útočníky. Tyto systémy se většinou používají pro speciální případy dvoubodových spojení (VPN) [10].

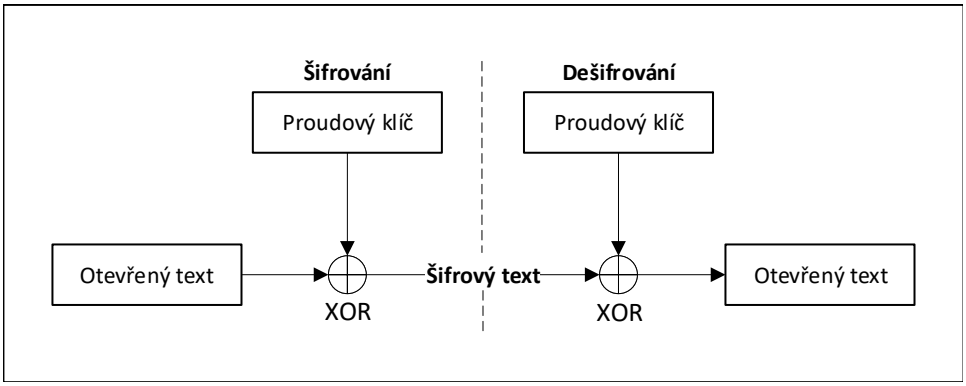
Blokové šifry

Zpráva je rozdělena do bloků a jsou šifrovány celé bloky dat najednou. Každý blok je šifrován pomocí stejného šifrovacího klíče a šifrování každého prvku závisí na ostatních prvcích bloku.

Proudové šifry

Klasická definice proudových šifer zní, že zpracovávají otevřený text znak po znaku, zatímco blokové šifry po blocích o několika znaků. Proudové šifry by tedy mohly

být chápány i jako blokové šifry s blokem délky jedna, avšak připomeňme, že tou podstatnou odlišností je, že u proudových šifer je každý tento blok zpracováván jiným způsobem, jinou substitucí. Princip jednoduché proudové šifry je vidět na Obr. 3.1.



Obr. 3.1: Princip proudové šifry (převzato z [11]).

Algoritmické proudové šifry

Děrné štítky byly časem nahrazeny kryptografickými algoritmy. Nejdříve se jednalo o mechanické šifrátory a později o elektronické šifrovací stroje. Heslo bylo šifrátory generováno a distribuovaly se pouze šifrovací klíče k nastavení těchto šifrátorů. Byl zaveden princip náhodně se měnícího inicializačního vektoru (IV). Díky tomuto principu nemusel být klíč měněn příliš často. IV byl vždy náhodně vybírán a posílán před šifrovým textem v otevřené podobě. Inicializační vektor nastavuje daný algoritmus (šifrátor) vždy do jiného počátečního stavu. Díky tomuto pak byla vždy generována jiná heslová posloupnost i při zachování stejného tajného klíče. IV zajišťuje různost hesla a šifrovací klíč zajišťuje utajenost [12].

3.2.2 Asymetrické kryptografické systémy

Asymetrické systémy jsou určeny k zabezpečení dat přenášených po veřejné síti. U těchto systémů se šifrovací klíče neshodují a zpravidla je jeden z klíčů veřejný a druhý tajný.

Veřejným klíčem je šifrovací klíč K_E

Na vysílací straně se použije veřejný klíč, který zašifruje data takovým způsobem, že znalost tohoto klíče nemá při dešifrování praktický význam, protože dešifrování

podle něj je časově a technicky mimořádně náročné. Strana příjemce má pro dešifrování jiný klíč, který vede k dešifrování podstatně rychleji. Tento klíč však musí být utajený. Tato metoda se využívá k zajištění důvěrnosti a integritě dat.

Veřejným klíčem je dešifrovací klíč K_D

Na vysílací straně se použije utajený klíč a k dešifrování slouží klíč veřejný. Tento způsob použití klíčů vytvoří autorův digitální podpis, který si může kdokoliv veřejným klíčem ověřit. Tento způsob zajišťuje autentičnost dat [10].

3.3 Výběr šifer

Výběr vhodné šifry je klíčový pro správné fungování zabezpečení na daném zařízení. Jelikož koncová zařízení Sigfox sítě by měla být co nejvíce energeticky úsporná, tak hlavním parametrem při výběru šifry je rychlost výpočtu. Vybrali jsme tedy několik šifer, které jsou rychlé a dobře se dají implementovat v softwaru. Následuje stručný popis uvažovaných šifer a problémů, které je potřeba vyřešit, aby šifra mohla fungovat.

One-Time Pad

One-Time Pad (OTP), jinak známá jako Vernamova šifra, se od ostatních šifer odlišuje hlavně tím, že používá jen operaci exkluzivního součtu XOR a díky tomu je také rychlejší než ostatní [13]. Jedná se o klasickou šifru. Pro správnou funkci šifry je potřeba použít unikátní klíč pro každou zprávu. Potřebuje tudíž generátor dokonale náhodných klíčů nebo alespoň velikou databázi předgenerovaných klíčů.

Advanced Encryption Standard

Šifra AES je velice známá a široce nasazovaná symetrická bloková šifra. Pro šifrování a dešifrování tedy využívá stejný klíč a data rozděluje do bloků o délce 128 bitů. To je první problém, který je potřeba vyřešit. Sigfox zprávy mají maximální délku payloadu stanovenou na 96 bitů, takže nejsme schopni odeslat celý blok šifrového textu. Možným řešením by bylo použít AES v proudovém režimu jako je např. Counter (CTR), Cipher Feedback (CFB), Output Feedback (OFB). Pro šifrování i následné dešifrování je také zapotřebí inicializačního vektoru (IV), který by měl být unikátní pro každou zprávu. Je tedy potřeba vyřešit jakým způsobem přenést IV na stranu příjemce, aby bylo možné zprávy dešifrovat [14].

Salsa20

Salsa20 je proudová šifra. Její šifrovací funkce používá tři jednoduché operace, kterými jsou bitový součet, exklusivní součet a bitový posun. Pro svou funkci potřebuje podobně jako AES společný klíč a pro každou zprávu unikátní IV. Ten však nemusí být zcela náhodný, ale skládá se z konstanty a pořadového čísla z čítače. Díky tomuto nemusíme přenášet pro každou zprávu celý IV, ale jen pořadové číslo z čítače [15].

ChaCha20

ChaCha20 je proudová šifra, která svojí funkcí vychází z šifry Salsa20. Je však rychlejší než Salsa20, čehož dosahuje jiným uspořádáním hodnot v matici, s kterou pak provádí jednoduché operace v jiném pořadí, než tomu tak je u Salsa20. Potřebné vstupní hodnoty jsou stejné. Je tudíž pořád potřeba vyřešit problém s přenesením IV na stranu příjemce [16].

Sosemanuk

Sosemanuk je také proudová šifra, která vychází z proudové šifry SNOW a blokové šifry Serpent. Délka klíče se může lišit v rozsahu od 128 do 256 bitů, avšak zaručená bezpečnost je pouze 128 bitů i při použití delšího klíče. Šifra také používá IV (128 bitů), které by bylo potřeba nějakým způsobem doručit na stranu příjemce [17].

Panama

Panama je kryptografický modul, který může být použit jako hašovací funkce nebo jako proudová šifra. Je navrhnut tak, aby byl co nejefektivnější při softwarové implementaci na 32-bitových systémech. Dosahuje vysoké rychlosti šifrování za cenu toho, že má velkou výpočetní režii v každém kole šifrovací funkce. To má za následek relativně dlouhou dobu inicializace. Příliš se nehodí v aplikacích, kde je rychlost klíčová a mělo by se při použití vyhnout časté resynchronizaci [18].

3.3.1 Shrnutí výběru

Hlavním zdrojem informací, který rozhodoval o výběru, byl Crypto++ 6.0.0 Benchmark [19], který navzájem srovnává nepřeberné množství běžně používaných algoritmů z hlediska rychlosti jejich fungování. Rychlost algoritmu je závislá na jeho optimalizaci a především na rychlosti samotného procesoru, na kterém jsou prováděna měření. Tento benchmark byl změřen na procesoru Intel Core-i5 Skylake s taktovací frekvencí 2,7 GHz [19]. Očekáváme, že rychlosti těchto algoritmů by

na procesoru, na který budeme sami implementovat vybrané šifry, budou znatelně nižší, protože námi zvolený procesor bude pracovat s taktovací frekvencí maximálně 16 MHz, tedy dosahuje menšího výkonu, a také použijeme jinak optimalizované algoritmy. Tento benchmark tedy slouží jen k porovnání rychlostí algoritmů mezi sebou.

Naměřené hodnoty z tohoto benchmarku pro uvažované šifry naleznete v Tabulce 3.1. Výjimkou je šifra OTP, která se v benchmark srovnání nenachází. Jelikož šifra k šifrování používá jen operaci exkluzivního součtu XOR, tak je zajištěno, že je potřeba menšího počtu cyklů potřebných k zašifrování dat než-li u ostatních šifer, které používají kombinaci několika operací [13].

Tabulka 3.1 obsahuje hodnoty kolik hodinových cyklů procesoru je potřeba k zašifrování jednoho bajtu, dvanácti bajtů, přípravy klíčů s IV a v posledním sloupci obsahuje hodnoty kolik cyklů je potřeba k přípravě a zašifrování dvanácti bajtů dohromady.

Tab. 3.1: Shrnutí výběru uvažovaných šifer (převzato z [19]).

Název šifry	Počet bitů klíče	Cyklů na bajt	Cyklů na 12 bajtů	Cyklů k přípravě klíče a IV	Cyklů k přípravě a zašifrování 12 bajtů
ChaCha20	256	5,16	61,92	252	313,92
Salsa20	256	2,48	29,76	372	401,76
AES CTR	128	0,57	6,84	598	604,84
Sosemanuk	128	1,48	17,76	1049	1066,76
Panama	256	1,36	16,32	1803	1819,32
Poznámka: Vychází z Crypto++ Benchmark. Měřeno na procesoru Intel Core-i5 Skylake 2,7 GHz.					

Pro další použití a implementaci jsme se rozhodli použít šifry AES CTR, ChaCha20 a OTP. Při šifrách ChaCha20 a Salsa20 jsme se rozhodli implementovat pouze jednu z nich, protože jsou si natolik podobné, že by následné porovnávání jejich implementací nemělo větší význam. Rozhodli jsme se pro ChaCha20, protože je rychlejší pro přípravu a zašifrování 12 bajtových zpráv. Dále budeme implementovat AES šifru v režimu CTR, která bude sloužit jako dobrý srovnávací příklad, jelikož je tato šifra široce nasazována a známa. Pro režim CTR jsme rozhodli na základě faktu, že nepotřebuje zcela náhodné IV, ale jen konstantu a hodnotu počítadla. To velice usnadňuje přenos IV na stranu příjemce. Jako poslední šifru jsme vybrali OTP z důvodu její rychlosti a jednoduchosti. Je však potřeba vyřešit problém s generováním klíčů.

4 POPIS VYBRANÝCH ŠIFER

Tato kapitola obsahuje podrobnější popis šifer AES, ChaCha20 a OTP. Tyto šifry byly vybrány v kapitole 3.3.

4.1 Advanced Encryption Standard

Advanced Encryption Standard (AES), jinak nazývaný také Rijndael (podle autorů Joana Daemena a Vincenta Rijmena), je kryptosystém nahrazující dříve používaný 3DES. Jedná se o symetrickou blokovou šifru dnes používanou např. pro bezdrátové Wi-Fi sítě se zabezpečením WPA2. Hlavní výhodou AES oproti jeho předchůdci je bezpečnost a rychlost, která nám umožnuje rychle šifrovat velké objemy dat. Prolomení nejbezpečnější verze AES hrubou silou by dnešním počítačům zabralo průměrně 149 biliónů let a museli by vyzkoušet přibližně $1,1^{78}$ kombinací [20]. Mezi hlavní vlastnosti AES, mimo vysoké rychlosti a odolnosti proti útokům hrubou silou, patří také nízké nároky na RAM což umožňuje implementovat AES do níkobitových mikrokontrolerů a zařízení s nízkým výpočetním výkonem obecně [12].

AES je symetrický blokový kryptosystém z čehož vyplývá, že využívá pro šifrování a dešifrování stejný klíč pro data s pevně stanovenou délkou bloku. AES má velikost bloků stanovenou na 128 bitů a velikost klíče je 128, 192 nebo 256 bitů. Pro naše použití je nejvhodnější AES-128 s délkou klíče právě 128 bitů.

Základní kroky algoritmu:

- Expanze klíče:
  - Dochází k odvození podklíčů z klíče šifry.
- Inicializace:
  - Provede se zkombinování bloku dat s podklíčem.
- Iterace:
  - Dojde k nahrazení každého bajtu jiným bajtem, prohození řádků, prohození sloupců a opětovnému zkombinování s podklíčem.
- Závěrečná část:
  - Dojde k nahrazení každého bajtu jiným bajtem,prohození řádků a opětovnému zkombinování s podklíčem (neprovádí se prohození sloupců).

Režimy šifrování

Kryptosystém AES není jen jeden šifrovací algoritmus. Může pracovat v několika režimech. Nejjednodušší z režimů je Electronic Codebook (ECB), kde je zpráva rozdělena do bloků a každý blok je šifrován odděleně. Pro zašifrování i dešifrování je

potřeba jen jeden klíč a v tom je právě slabina tohoto režimu. Pokud jsou šifrována stejná data, tak i výsledný šifrový text má stejnou podobu a tím odhalujeme určitou podobnost mezi zprávami. Z těchto důvodů se tento režim moc nepoužívá a raději se volí jiné složitější režimy.

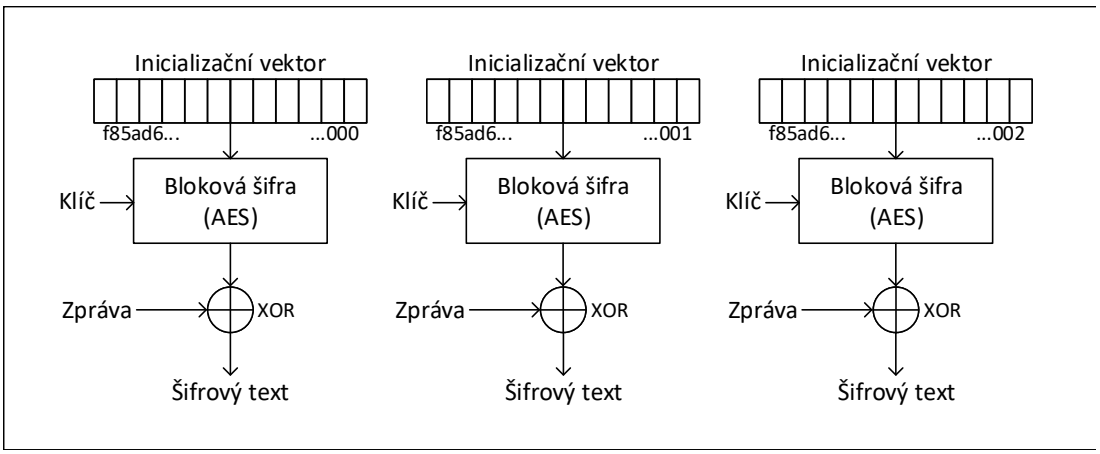
Další z režimů je Cipher Block Chaining (CBC), v kterém se již bloky nešifrují samostatně, ale k zašifrování prvního bloku je potřeba nejen klíč, ale také inicializační vektor (IV). Pro zašifrování dalších bloků je potřeba předchozí zašifrovaný blok a klíč. To nám umožňuje dešifrovat jakýkoliv blok šifrového textu nezávisle na prvním bloku a také díky tomu můžeme dešifrovat celý šifrový text bez IV s výjimkou prvního bloku. Dešifrování probíhá inverzně k šifrování. Tento režim je již bezpečnější než ECB, ale je potřeba k straně, kde probíhá dešifrování, dopravit IV, kterým se daná zpráva zašifrovala [21].

U blokových režimů ovšem nastává problém, že je zapotřebí odeslat minimálně jeden celý blok, aby bylo možné na straně příjemce zprávu dešifrovat. Nemůžeme poslat jen část bloku, protože informace o vstupním textu je obsažena v celém bloku šifrového textu. Jelikož minimální délka bloku u AES šifry je stanovena na 16 bajtů, tak ho není možné přenést v jedné Sigfox zprávě, která má maximální délku uživatelských dat stanovenou na 12 bajtů. Z tohoto důvodu není možné použít jakýkoliv blokový režim. Existují však i proudové režimy, které umožňují vstupní data o délce 12 bajtů zašifrovat do 12 bajtů dlouhého šifrového textu. Jedním z představitelů proudových režimů je režim Counter (CTR).

Režim Counter je jeden z nejjednodušších proudových režimů, který umožňuje použít blokovou šifru jako šifru proudovou. Výsledný šifrový text poté nemá délku o velikosti násobků velikosti jednoho bloku (n-bloků), ale jeho délka je volitelná, resp. stejná jako velikost vstupních dat. Šifra AES při práci v tomto režimu potřebuje k šifrování jeden náhodný klíč o velikosti 16 bajtů a IV také o velikosti 16 bajtů [22].

Základním pravidlem pro správné fungování CTR režimu je, že se žádná zpráva nesmí zašifrovat vícekrát se stejným klíčem a IV [22]. Proto se IV skládá ze dvou částí - z části konstanty (tzv. nonce) a části pro hodnoty počítadla. Právě počítadlo zajišťuje, že se žádná zpráva nezašifruje víckrát se stejným klíčem a IV. Část pro hodnoty počítadla se s každou zašifrovanou zprávou zvyšuje o jedno a tudíž, aby výše zmíněné pravidlo bylo splněno, tak se počítadlo nesmí nikdy resetovat, resp. část počítadla musí být dost velká, aby se nikdy zcela nenaplnila a nepřetekla zpět na hodnotu, která již byla dříve použita při šifrování.

Samotná funkce režimu CTR je znázorněna na Obrázku 4.1 a spočívá v tom, že se blokovou šifrou (AES-128) zašifruje klíč společně s IV. Poté se provede exklusivní součet XOR mezi výsledkem blokové šifry a zprávou, kterou chceme zašifrovat, čímž vznikne výsledný šifrový text. Pro dešifrování zpráv je postup obdobný, pouze se exklusivní součet provádí mezi výsledkem blokové šifry a šifrovým textem, čímž získáme zpět původní nezašifrovanou zprávu.



Obr. 4.1: Princip režimu Counter (Převzato z [22]).

4.2 ChaCha

ChaCha je varianta šifry Salsa20, která má za cíl zlepšit difúzi šifry bez zpomalení šifrování. Jedná se o proudovou šifru, která byla představena D. J. Bernsteinem v roce 2008 jako kandidát v projektu eSTREAM [16]. V současnosti ji používá společnost Google spolu s MAC algoritmem Poly1305 od stejného autora jako náhradu za RC4 v TLS [23]. Dále se také používá v generátoru náhodných čísel arc4random na svobodných operačních systémech FreeBSD, OpenBSD a NetBSD [24][25][26] a také v hashovací funkci BLAKE, která byla jedním z pěti finalistů při výběru NIST SHA-3.

Šifra pracuje s 32 bitovými slovy. Jako vstup vyžaduje 256 bitový klíč $k = (k_0, k_1, ..., k_7)$ a 64 bitovou konstantu $v = (v_0, v_1)$ a 64 bitové počítadlo $t = (t_0, t_1)$. Konstanta společně s počítadlem tvoří IV, podobně jako tomu je u proudového režimu CTR. Šifrovací funkce se chová jako matice 4×4 tvořená

z 32 bitových slov uspořádaných následovně:

$$X = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & t_1 & v_0 & v_1 \end{pmatrix}. \tag{4.1}$$

Hodnoty c_i jsou předdefinované konstanty. Existuje také varianta šifry se 128 bitovým klíčem, ale je doporučeno používat vždy klíč o velikosti 256 bitů. ChaCha, podobně jako Salsa20, používá k šifrování 4 součty modulo 2^{32} (značeno $+$), 4 operace XOR (značeno $\oplus$) a 4 bitové rotace směrem k významnějším bitům (značeno $\ll$) čímž transformuje hodnoty (x_0, x_1, x_2, x_3) na hodnoty (z_0, z_1, z_2, z_3) následujícím způsobem:

$$\begin{aligned} b_0 &= x_0 + x_1, & b_3 &= (x_3 \oplus b_0) \ll 16 \\ b_2 &= x_2 + b_3, & b_1 &= (x_1 \oplus b_2) \ll 12 \\ z_0 &= b_0 + b_1, & z_3 &= (b_3 \oplus z_0) \ll 8 \\ z_2 &= b_2 + z_3, & z_1 &= (b_1 \oplus z_2) \ll 7\ . \end{aligned} \tag{4.2}$$

Průběh těchto operací se nazývá čtvrtkola (z anglického quarterround) a maximálně se provádí dvacetkrát, z toho vyplývá název ChaCha20. Pro aplikace, kde se upřednostňuje rychlost šifrování před bezpečností, je možné použít varianty se sníženým počtem čtvrtkol - ChaCha12 a ChaCha8 [27].

4.3 One-Time Pad

Šifra One-Time pad (známá také jako Vernamova šifra) je jednoduchý, avšak účinný šifrovací algoritmus. Její princip spočívá v posunu každého znaku zprávy o určitý počet míst v abecedě, podobně jako tomu je v Caesarově šifře. Rozdíl je v tom, že zde se každý znak posouvá o náhodný počet míst. Prakticky dojde k náhradě zcela náhodnými znaky a na tomto faktu je založen důkaz o nerozluštitelnosti této šifry (absolutně bezpečná šifra). Posloupnost čísel o kolik se mají dané znaky posunout určuje tajný klíč, který by měl znát jen odesílatel a příjemce. Velice důležité je zajištění bezpečného přenosu tajného klíče k příjemci.

Vernamova šifra se dříve používala a dosud ještě používá pro zabezpečení nejdůležitějších spojů, kde je nutné mít zajištěnou stoprocentní bezpečnost (diplomatické spoje). Nevýhodou je nutnost dopravit klíč k oběma stranám komunikačního kanálu. Dříve se používali děrné pásky, které se přenášeli v diplomatických zavazadlech. Aby se nemusel pro každou zprávu přenášet děrný pásek, tak docházelo k jejich opakovanému používání, a tedy i k možnému úniku tajných informací [10].

Podmínky spolehlivosti

- Klíč je stejně dlouhý jako přenášená zpráva:
  - Kratší klíč umožňuje snadnější prolomení hrubou silou.
- Klíč je dokonale náhodný:
  - Nelze použít počítačové pseudonáhodné generátory. Nejvhodnější je užití hardwarových generátorů náhodných čísel.
- Klíč se smí použít pouze jednou:
  - Opakovaný klíč již není klíčem náhodným. Dvě zprávy zašifrované stejným klíčem je snadné rozluštit.
  - V binární variantě je toto pravidlo velice důležité, protože pro operaci XOR (z anglického exclusive or; značeno jako $\otimes$) platí:

$$C_1 \oplus C_2 = (A \oplus X) \oplus (B \oplus X) = A \otimes B \ , \tag{4.3}$$

    kde C_1, C_2 značí šifrové texty, A, B značí zprávy a X značí šifrovací klíč.

Důkaz bezpečnosti

Tato část přináší důkaz o dokonalé tajnosti šifry OTP [28].

Theorem 1. Pokud máme šifrovací funkci *Enc* a prostor pro zprávu M, tak *Enc* nad M je dokonale bezpečná pouze, pokud pro všechny zprávy $m \in M$ a šifrový text c platí pravděpodobnost $P[m|c] = P[m]$.

Důkaz. Pro všechny $M = \{0,1\}^n$, kde n značí počet zpráv, tak pro $m \in M$ a jakékoliv c platí:

$$P[m|c] = \frac{P[m \wedge c]}{P[c]} = \frac{P[c|m] \cdot P[m]}{P[c]} \ . \tag{4.4}$$

Pravděpodobnost šifrového textu $P[c]$ pro všechny zprávy v prostoru pro zprávy je suma:

$$P[c] = \sum_{m \in M} P[c|m] \cdot P[m] \ , \tag{4.5}$$

kde pro všechny m,c platí:

$$P[c|m] = P[c \oplus m] = 2^{-n} \ . \tag{4.6}$$

Sloučením rovnic 4.5 a 4.6 dostáváme:

$$P[c] = \sum_{m \in M} 2^{-n} \cdot P[m] = 2^{-n} \ . \tag{4.7}$$

Dosazením rovnic 4.6 a 4.7 do rovnice 4.4 dostáváme:

$$P[m|c] = \frac{2^{-n} \cdot P[m]}{2^{-n}} = P[m] \ . \tag{4.8}$$

□

Důkaz korektnosti

Následující důkaz potvrzuje, že příjemce opravdu obnoví původní text při dešifrování šifrového textu, který obdržel s použitím šifry OTP [29].

Theorem 1. Pokud máme šifrovací funkci

$$Enc(k, m) = k \oplus m \tag{4.9}$$

a dešifrovací funkci

$$Dec(k, c) = k \oplus c \ , \tag{4.10}$$

kde k je šifrovací klíč, m je zpráva a c je šifrový text, tak pro všechny $k, m \in \{0,1\}^n$, kde n značí délku k a m, platí že:

$$Dec(k, Enc(k, m)) = m \ . \tag{4.11}$$

Důkaz. Pokud do rovnice dosadíme definované vztahy pro Enc a Dec, poté vypočteme, tak pro všechny $k, m \in \{0,1\}^n$ dostáváme výsledek:

$$\begin{aligned} Dec(k, Enc(k, m)) &= Dec(k, k \oplus m) \\ &= k \oplus (k \oplus m) \\ &= (k \oplus k) \oplus m \\ &= 0^n \oplus m \\ &= m \ . \end{aligned} \tag{4.12}$$

□

Příklad

Šifrování následujícího textu m s klíčem k má za výsledek šifrový text c.

$$\begin{aligned} &00110100110110001111 \ m \\ \oplus\ &\underline{11101010011010001101 \ k} \\ &11011110101100000001 \ c \end{aligned}$$

Dešifrování c za použití stejného klíče k má za výsledek originální m.

$$\begin{aligned} &11011110101100000001 \ c \\ \oplus\ &\underline{11101010011010001101 \ k} \\ &00110100110110001111 \ m \end{aligned}$$

5 REALIZACE ZABEZPEČENÍ

V této kapitole je uveden příklad reálného nasazení zařízení v praxi a také jaký hardware byl použit včetně jeho zapojení. Dále jsou vysvětleny implementace jednotlivých šifer (rozložení IV, předání IV a klíče straně příjemce).

5.1 Příklad nasazení v praxi

Jedním z častých případu nasazení LPWAN technologií je měření teploty. Proto jsme se rozhodli toto nasazení uvést do praxe a zabezpečit ho pomocí našich implementací šifer AES, ChaCha a OTP. Zařízení má za úkol v pravidelných intervalech, které je možné libovolně nastavit, změřit teplotu okolí, zašifrovat tuto informaci o teplotě a takto zabezpečená data odeslat po Sigfox síti k příjemci. Jako příjemce si můžeme představit server, který data shromažďuje a skrze webové stránky předává informace k uživateli. Řešení serveru shromažďující informace není součástí bakalářské práce.

Aby zařízení mohlo fungovat na bateriový akumulátor co nejdéle, tak je nezbytné, aby se mikroprocesor a i ostatní části zařízení uspávali mezi intervaly měření. Právě mikroprocesor zajistí uspání všech součástí zařízení, ale pro pravidelné probouzení budeme potřebovat nejlépe energeticky úsporný obvod real-time clock (RTC). Ten jako jediný nepřechází do úplného spánku a v pravidelných intervalech probouzí mikroprocesor, který poté probouzí další částí zařízení ve chvíli kdy je jich zapotřebí.

Všechny 3 implementace jsou naprogramovány tak, aby po probuzení, způsobeném RTC obvodem, mikroprocesor odeslal příkaz k měření teploty do teploměru a také probudil Sigfox modulek. Po uplynutí doby potřebné k změření teploty si mikroprocesor vyžádá data od teploměru, které hned po přijetí zašifruje. Po zašifrování se zpráva odešle následovaná příkazem k uspání. Tento proces se dá opakovat po libovolně dlouhé době. Popis použitých hardwarových součástí naleznete v následující kapitole.

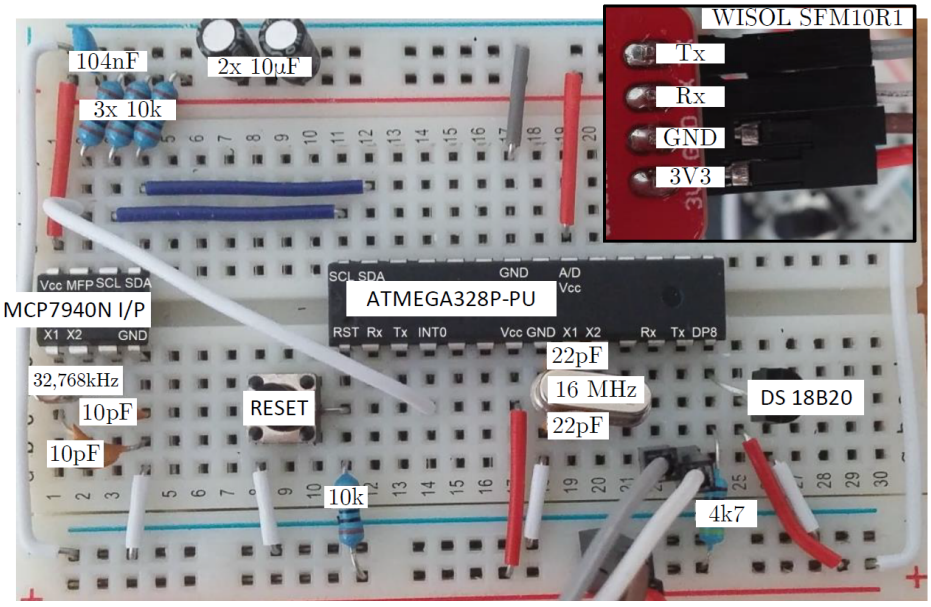
5.1.1 Použitý hardware

Hlavní součástkou, která provádí většinu výpočtů, je 8 bitový AVR mikroprocesor ATMEGA328P-PU. Tento mikroprocesor má minimální odběr až 0,1 µA (při 1 MHz; 1,8 V; 25 °C) a jeho maximální taktovací frekvence je 20 MHz [30]. Ze spánku ho je možné probudit jen restartem napájení nebo pomocí tzv. přerušení (interrupt). Ty spočívají v tom, že se na pinech mikroprocesoru INT0 nebo INT1 změní logická

hodnota signálu. Mikroprocesor pak na základě nastavení reaguje buď na náběžnou, nebo sestupnou hranu signálu a probudí se. Tento mikroprocesor se používá také v zařízeních Arduino UNO, na kterém by bylo možné naše implementace šifer zrealizovat, avšak UNO obsahuje mnoho vedlejších součástek, které nejsou přímo potřebné k běžnému chodu. Tyto nadbytečné součástky neustále odebírají určitý proud, proto jsme se rozhodli si vytvořit vlastní zapojení skládající se jen z nezbytných součástek. Schéma zapojení můžete nalézt v příloze A.

Další velice důležitou součástkou je RTC MCP7940N I/P, který má za úkol udržovat v paměti přesný čas a datum. Na základě předchozího nastavení umí tato součástka v daný čas alarmovat mikroprocesor skrze změnu signálu na svém MFP pinu. To se dá využít například pro pravidelné vykonávání nějaké činnosti nebo pro probouzení ze spánku. Tento RTC čip dosahuje minimálního odběru ve spánku až 1,4 µA.

Pro náš příklad nasazení v praxi potřebujeme také teploměr. Rozhodli jsme se použít teploměr DS18B20. Jedná se o digitální energeticky úsporný teploměr, který informaci o teplotě posílá skrze svůj DQ pin pomocí 1-Wire protokolu s rozlišením v rozsahu 9 až 12 bitů. Přímo úměrně závislá na nastaveném rozlišení je doba potřebná k změření teploty, která se pohybuje v rozsahu 93,75 až 750 ms. Minimální odebíraný proud ve spánku je až 0,75 µA.



Obr. 5.1: Fotografie zkušebního zapojení (převzato z [31]).

Nezbytnou součástí zařízení pracující v Sigfox síti je modulek, který zajišťuje vytvoření Sigfox rámce a jeho odeslání pomocí 5 dB antény do okolí k základnovým stanicím. Jeden z nejpoužívanějších je modulek osazený čipem WiSOL SFM10R1, který data přijímá přes sériovou komunikaci. Pracovní napětí tohoto čipu je v rozsahu od 1,8 do 3,6 V. Z důvodu tohoto omezení není možné napájet celé zařízení s napětím větším než 3,6 V. Při probouzení modulku je nutné příkaz k probuzení odeslat přibližně 55 ms před odesláním jiných příkazů. Tato doba je nezbytná k úplnému probuzení modulku a případné dřívěji odeslané příkazy modulek ignoruje.

Jako napájecí zdroj jsme vybrali lithiovou AA baterii SAFT LS14500 s napětím 3,6 V a kapacitou 2600 mAh. Tato kapacita by měla stačit k napájení zařízení po dobu několika let. Pro energetický náročnější nasazení je možné použít větší počet těchto baterii, ale s tím by také razantně vzrostla cena zařízení.

Pro případ kdy bychom chtěli zprávy šifrovat šifrou OTP, tak je nezbytné použít paměťový čip, který by byl schopen uložit všechny klíče potřebné pro šifrování po celou dobu životnosti zařízení. Vhodným příkladem by mohla být paměť AT27LV040A od firmy Microchip, která má velikost 4 Mb [32], což by mělo vystačit na dobu 6 let.

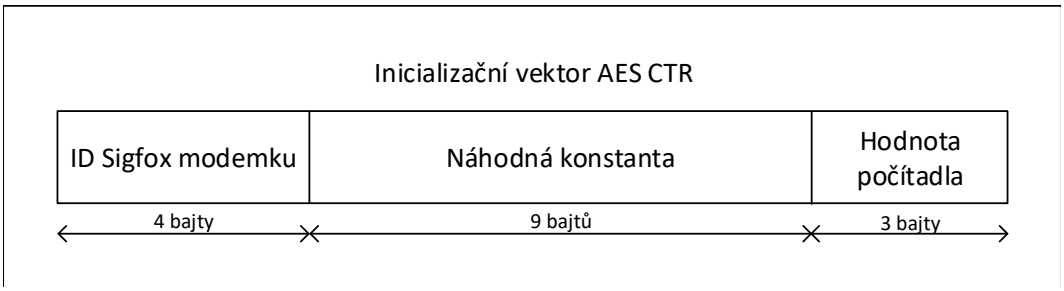
Co se týče ochrany citlivých dat potřebných k fungování šifer AES a ChaCha, tak tato data by byla uložena v paměti mikroprocesoru společně s programem, protože mikroprocesor disponuje dostatečně velkou vnitřní pamětí. Pro jejich zabezpečení mikroprocesor nabízí funkci "Programming Lock for Software Security", která zajišťuje jednoduchou ochranu celého firmwaru [30].

5.2 Realizace Advanced Encryption Standard

Jak již bylo řečeno v kapitole 3.3.1, tak jsme se rozhodli implementovat šifru AES pracující v režimu CTR. Režim CTR nám velice usnadňuje řešení problému s dopravením IV na stranu příjemce. Narozdíl od blokových režimů nemusí být IV náhodný, ale naopak musí obsahovat část konstanty a část vyhrazenou pro hodnoty počítadla. My jsme se rozhodli v naší implementaci vyhradit pro konstantu část o velikosti 13 bajtů, která se skládá z 4 bajtového ID Sigfox modemku a zbylých 9 bajtů obsahuje náhodně vygenerovaná čísla pro vnesení větší náhodnosti na vstup šifrování.

Druhá část IV, hodnota počítadla, má velikost 3 bajty. Takovéto rozložení IV nám dává možnost zašifrovat až $2^{24} = 16\,777\,216$ zpráv. Teprve až po překročení tohoto počtu by počítadlo začalo počítat znovu od původní hodnoty a tím by byla porušena základní podmínka bezpečnosti CTR režimu. Jelikož se životnost LPWAN

zařízení pohybuje kolem pěti let a za tuto dobu by zařízení mohlo maximálně odeslat $140 \times 365 \times 5 = 255\,500$ zpráv, tak nám takto veliká část vyhrazená pro hodnoty počítadla vystačí. Pro ještě větší vnesení náhodnosti do šifrování bychom mohli z části pro počítadlo ubrat až 6 bitů, které by byly nahrazeny náhodnými hodnotami. Po ubrání by počítadlo dosahovalo maximálně hodnoty $2^{18} = 262\,144$, ale toto řešení vyžaduje složitější úpravy programového řešení, jelikož aktuální řešení pracuje s celými bajty. Rozložení IV můžete vidět na Obr. 5.2.



Obr. 5.2: Rozložení IV šifry AES-128 v režimu CTR.

Jelikož se tedy IV skládá z konstanty a hodnoty počítadla, která se inkrementuje s každou odeslanou zprávou, tak je IV možné zjistit i na straně příjemce pomocí sekvenčního číslování, které je společně v Sigfox rámcích vždy posíláno. Konstanta by se před nasazením zařízení do praxe uložila do databáze a podle ID Sigfox modemku, který se také odesílá v Sigfox rámcích, by se tato konstantní část z databáze načetla a použila při dešifrování na straně příjemce. Šifrovací klíč by se na straně příjemce synchronizoval obdobným způsobem jako konstantní část IV. Samotná šifrovací, resp. dešifrovací funkce byla převzata z Arduino Cryptography Library [33].

5.3 Realizace ChaCha

Při použití šifry ChaCha máme na výběr dvě varianty, které využívají stejné šifrovací postupy, ale liší se v rozložení IV. První možnost je použít šifru tak, jak ji původně vytvořil D. J. Bernstein. Ta tedy k šifrování vyžaduje 254 bitový náhodný klíč a IV, který se skládá z 8 bajtové konstanty a 8 bajtové části pro počítadlo. Druhou variantu představila skupina Internet Research Task Force (IRTF) [34], která se od původní liší hlavně v tom, že se IV skládá z 12 bajtové konstanty a jen 4 bajtové části pro počítadlo. My se rozhodli použít variantu popsanou skupinou IRTF, protože nám vyhovuje již délka počítadla 3 bajty (popsáno v předešlé kapitole 5.2).

Dopravení IV a šifrovacího klíče na stranu příjemce zajišťuje vzájemné sesynchronizování části počítadla IV se sekvenčním číslováním Sigfox zpráv, jak je taktéž popsáno v předešlé kapitole 5.2. Samotný šifrovací algoritmus jsme převzali z Arduino Cryptography Library [33], která umožňuje obě možné varianty rozložení IV.

5.4 Realizace One-Time Pad

Jelikož šifra OTP pro zašifrování jednoho bitu zprávy využívá pouze operaci exklusivního součtu XOR, tak je její realizace velice jednoduchá. Popis šifrovací funkce, kterou jsme použili v naší implementaci naleznete níže v Algoritmu 1.

Algorithm 1 One-Time Pad šifrování

Require: Jednorázový klíč $k \in \{0,1\}^n$, zprávu $m \in \{0,1\}^n$, kde n je velikost k, m
Ensure: Šifrový text $c \in \{0,1\}^n$

```
for (i ← 0) to n − 1 do
    c_i ← m_i ⊕ k_i
end for
return c
```

Jedním z klíčových problémů, který je potřeba vyřešit při implementaci šifry OTP, je dopravení šifrovacího klíče na stranu příjemce. Jedním z řešení je vytvořit předgenerovanou databázi klíčů, kterou by disponovalo jak koncové zařízení, tak i příjemce. Ta by obsahovala klíče pro všechny zprávy, které je zařízení schopné odeslat za celou dobu jeho životnosti. Pro dobu životnosti 5 let by bylo potřeba uložit 255 500 klíčů, které by zabírali přibližně 3 MB paměti. Takové množství dat už není schopen mikroprocesor uložit ve své vnitřní paměti a bylo by tedy zapotřebí externího paměťového čipu, jako je například paměť AT27LV040A (odběr ve spánku pod 0,1 µA).

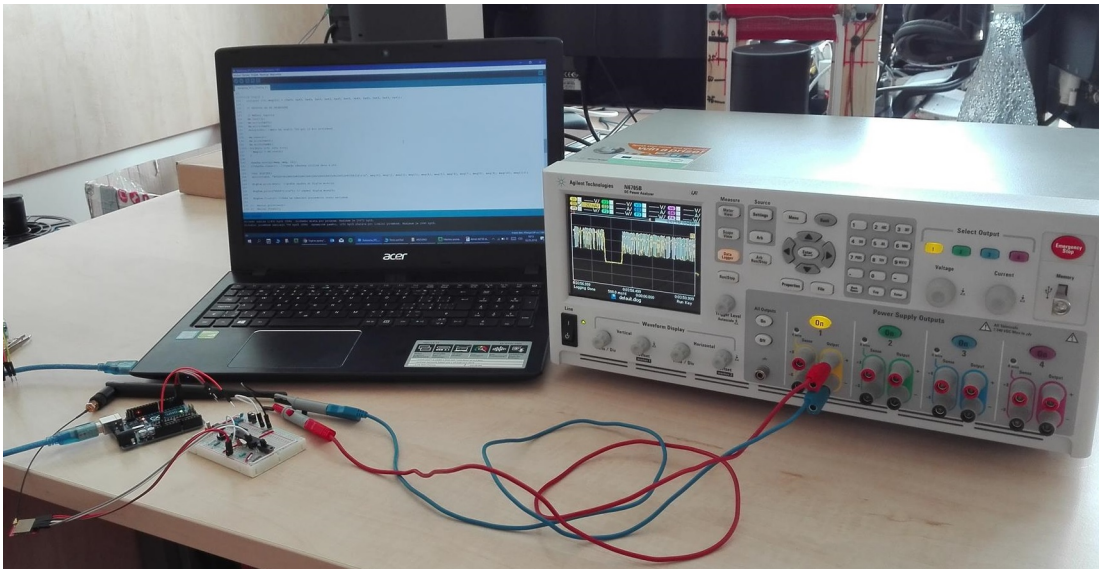
Výběr klíčů při dešifrování je nutné synchronizovat s klíči, které byly použity při zašifrování zpráv. Tuto synchronizaci opět zajistí sekvenční číslování. Databáze by měla ke každému klíči přiřazené číslo (např. číslo řádku, na kterém se klíč nachází v databázi), které by odpovídalo aktuálnímu sekvenčnímu číslu.

6 MĚŘENÍ

Měření energetické náročnosti zařízení

Cílem tohoto měření bylo zjistit jak hodně energeticky náročná jsou jednotlivá řešení šifer. Byl tedy změřen celý průběh odběru proudu od probuzení zařízení do jeho opětovného uspání, což zahrnuje část práce mikroprocesoru a část odesílání zprávy Sigfox modulkem. To bylo provedeno pro případ bez šifrování a případy s šifrováním šifer AES, ChaCha a OTP. Zmiňované případy byly změřeny s taktovací frekvencí mikroprocesoru 16 MHz. Měření bylo zrealizováno za pomoci přístroje Agilent Technologies N6705B DC Power Analyzer při teplotě okolí 25 °C. Měřící pracoviště můžete vidět na Obrázku 6.1.

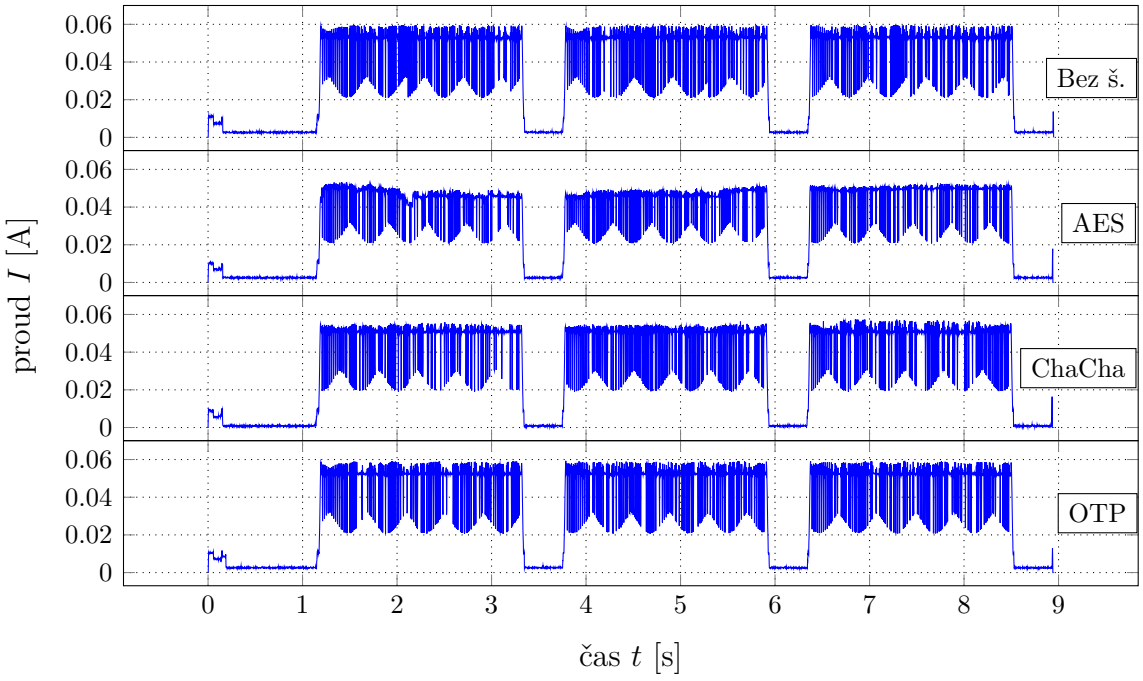
Očekávanými výsledky tohoto měření jsou hodnoty odběru proudu zařízení v závislosti na čase pro případ bez šifrování a případ s šifrováním AES, ChaCha a OTP, které budou vloženy do grafů. Tyto grafy jsou velice důležité pro zjištění jak moc šifry ovlivňují odběr proudu v porovnání s celkovým odběrem zařízení. Dalším výsledkem je také průměrná hodnota proudu odebíraná v době spánku zařízení, která by se měla pohybovat v řádu jednotek mikroampér.



Obr. 6.1: Fotografie měřícího pracoviště.

Výsledky

Z Obrázku 6.2 je zřejmé, že mezi průběhy odběru jednotlivých šifer není znatelný rozdíl. I při porovnání šifer s průběhem bez šifrování se mezi sebou liší jen minimálně, což dokazuje, že použité šifrování má minimální vliv na výdrž baterie zařízení, resp. na životnost zařízení. Všechny průběhy se skládají z dvou hlavních částí. První je část práce mikroprocesoru (první impuls o velikosti přibližně 10 mA). Ten nastává vždy po probuzení ze spánku a je v něm také obsaženo šifrování. Následuje přibližně sekundu dlouhá část menšího odběru (přibližně 2 mA), kde dochází ke zpracování přijaté zprávy Sigfox modulkem a přípravě Sigfox rámce k odeslání. Druhá hlavní část se skládá z tří kolísajících, dominantních pulzů, které dosahují odběru až 60 mA. Ty jsou způsobeny odesíláním Sigfox zpráv do okolí (opakované odeslání třikrát po sobě). V režimu spánku celé zařízení odebírá v průměru proud o velikosti 9 µA.



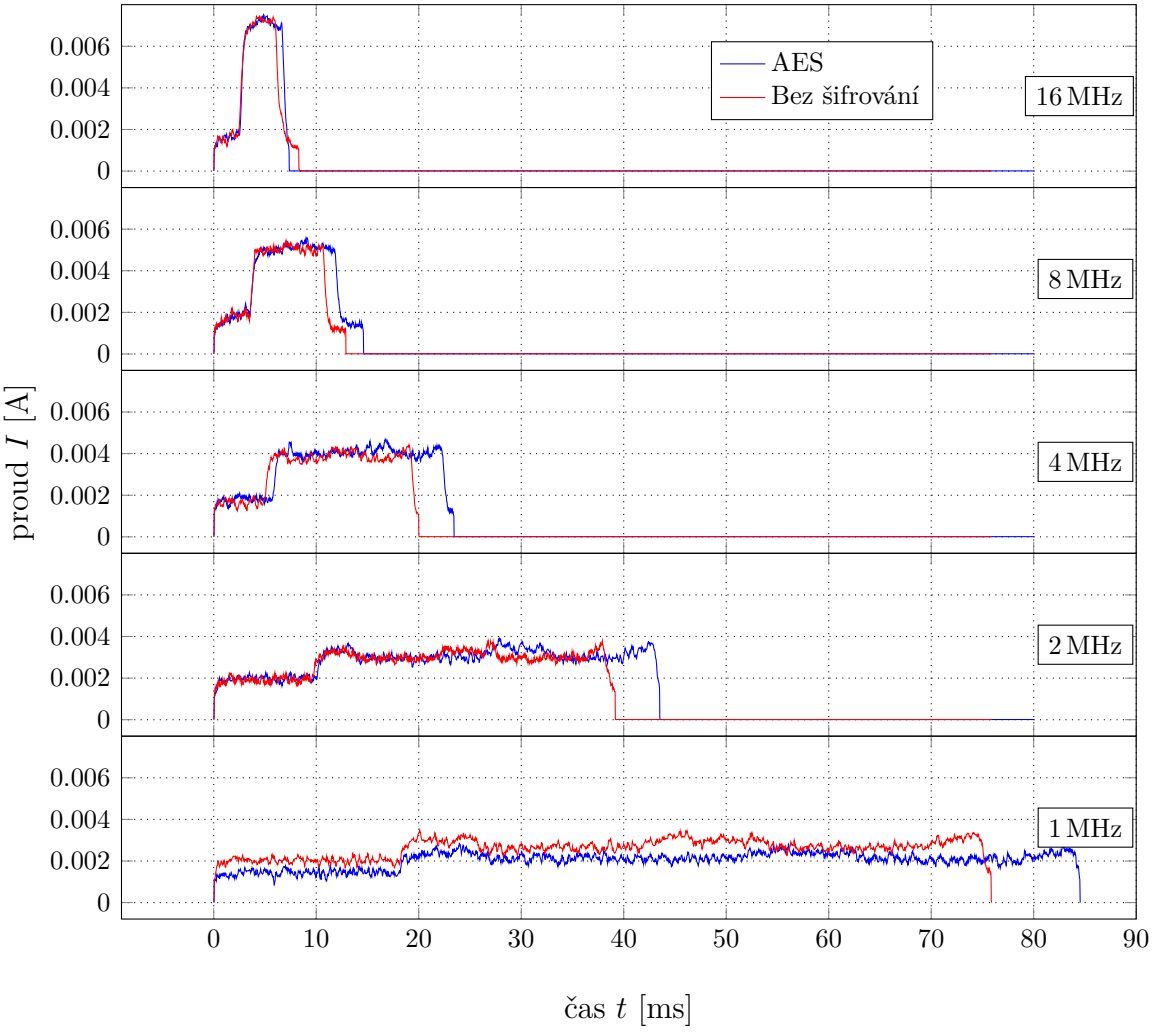
Obr. 6.2: Grafy odběrů šifer v porovnání s průběhem bez šifrování (16 MHz).

Měření energetické náročnosti šifrování

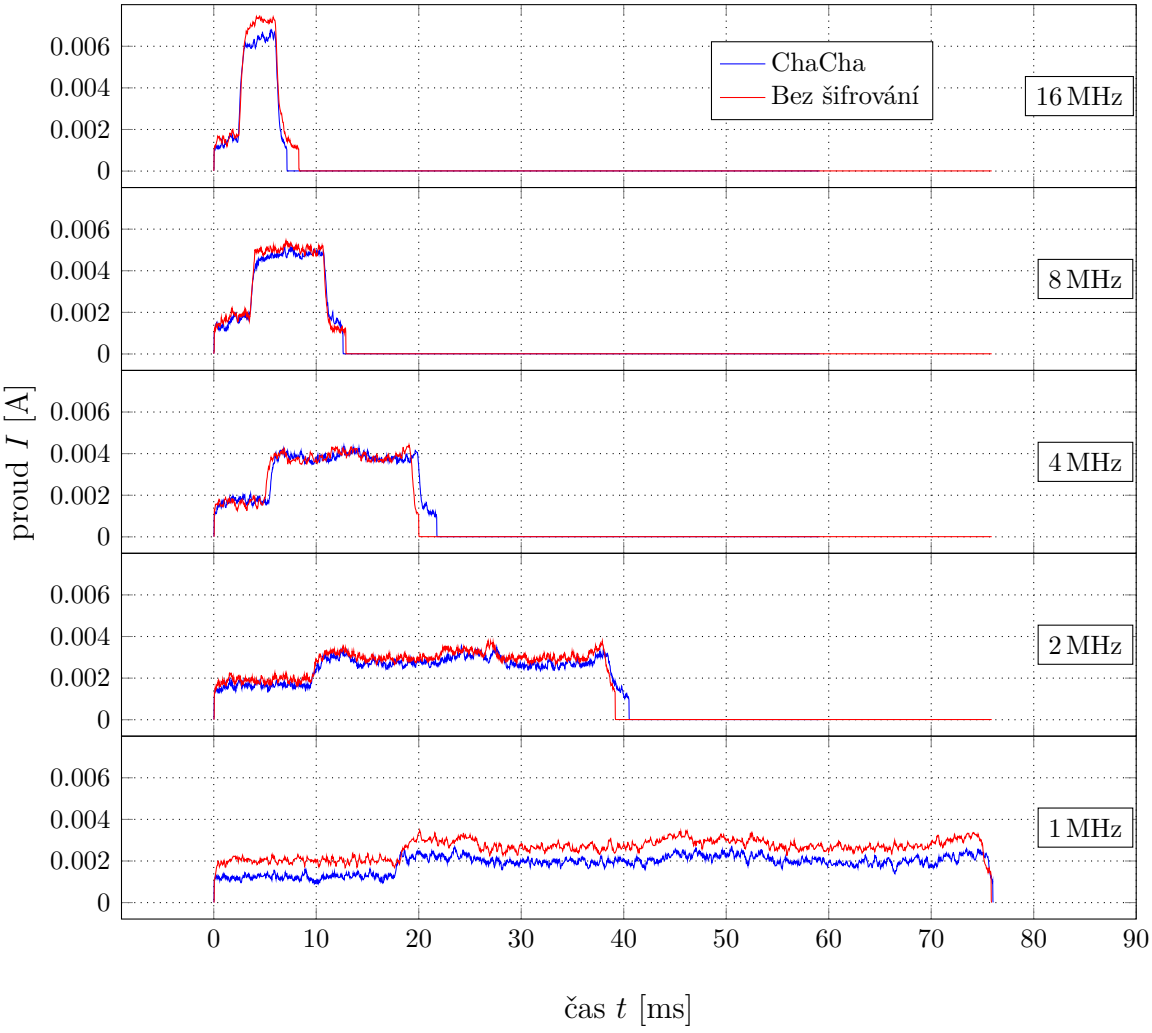
Toto měření mělo za cíl zjistit průběh odebíraného proudu zařízení při šifrování AES, ChaCha a OTP pro různé taktovací frekvence mikroprocesoru. V předchozím měření jsme měřili průběhy odběru celého procesu zpracování a odeslání zprávy. Zde jsme se však zaměřili jen na část práce mikroprocesoru bez komunikace s teploměrem či Sigfox modulkem. Mikroprocesor se pro toto měření jen probudil ze spánku, zašifroval statickou zprávu a opět se uspal. Probouzení pořád obstarával obvod RTC, jak již bylo popsáno v kapitole *5 Realizace zabezpečení*. Měřící přístroj Agilent N6705B umožňoval měření proudu přibližně nad hodnotu 1 mA, což nám dovolovalo měřit s taktovacími frekvencemi od 1 MHz do 16 MHz. Při nižších taktovacích frekvencích jíž byl odběr tak nízký, že výsledné průběhy nebyly čitelné.

Výsledky

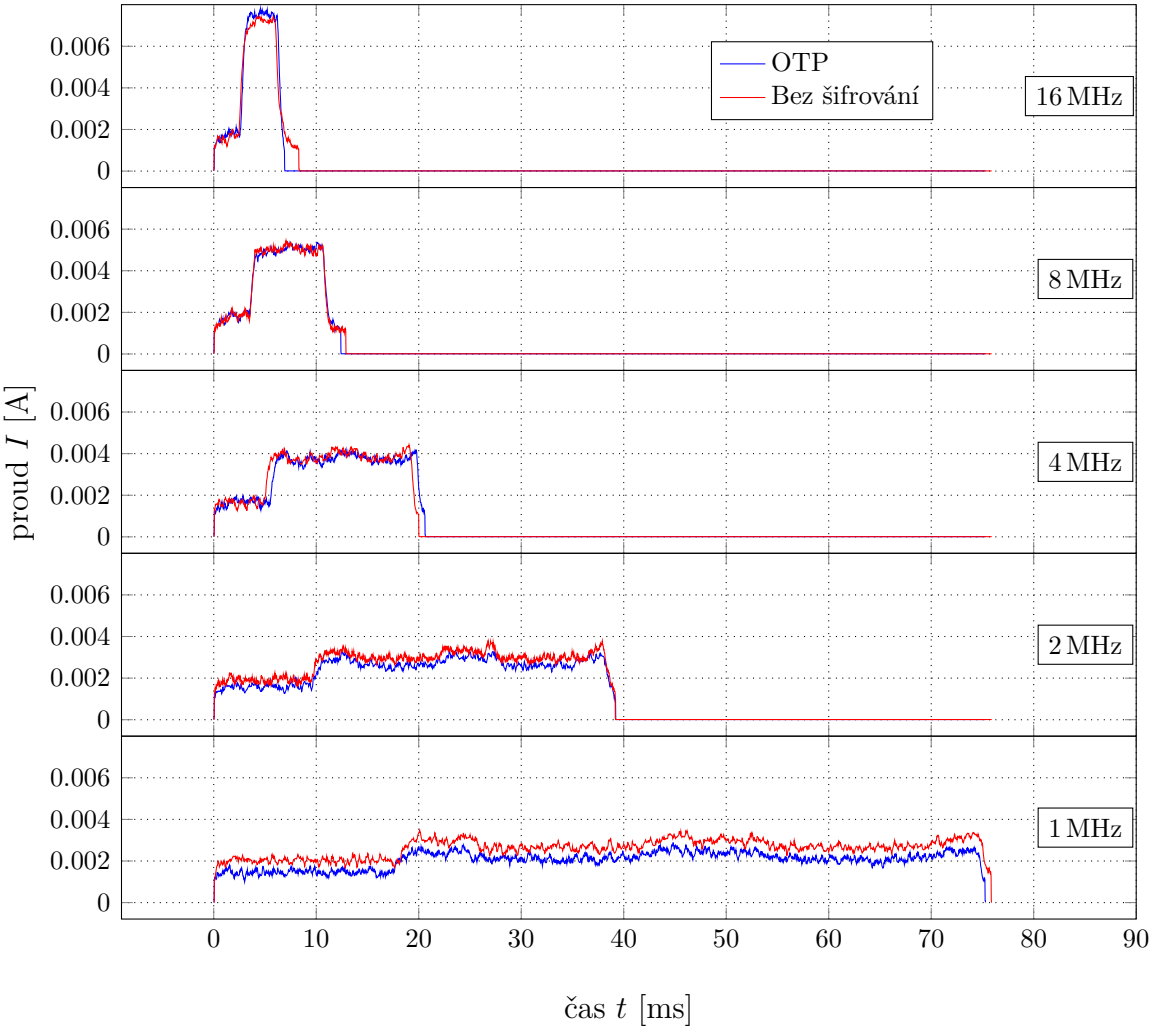
Výsledky měření odběru šifry AES můžete vidět na Obr. 6.3. Z grafu je vidět, že přidáním šifrování se aktuální hodnoty proudu razantně nezměnily, ale prodloužila se doba potřebná k vykonání instrukcí programu. Tato doba s klesající taktovací frekvencí roste lineárně (s poloviční frekvencí se doba zvětší přibližně dvojnásobně). Hodnoty proudu s klesající taktovací frekvencí klesají nelineárně, což má za následek zvětšování odebraného proudu potřebného k zašifrování jedné zprávy. Jinak řečeno, výsledná hodnota integrálu $\int_a^b I dt$, kde a je čas probuzení a b je čas uspání, s klesající frekvencí roste. Dle měření by tedy mělo být úspornější použití vyšších taktovacích frekvencí. Dále si můžeme povšimnout, že u šifry ChaCha (Obr. 6.4) i OTP (Obr. 6.5) je doba šifrování minimální v porovnání s dobou potřebnou v případě šifry AES. Měření šifry OTP však nezahrnovalo načtení klíče z databáze ani odběr samotného paměťového čipu, jehož odběr je přibližně 10 µA (pro AT27LV040A) [32]. U průběhů s taktovací frekvencí 1 MHz je přítomna znatelná chyba v podobě nižších hodnot proudu při šifrování oproti průběhu bez šifrování. S větším zatížením mikroprocesoru by měl být odebíraný proud vyšší. Tato chyba je nejspíše způsobena měřícím přístrojem, který při měření nižších hodnot proudu posouvá nulovou hodnotu proudu až do záporných hodnot. V důsledku toho jsou pak průběhy v grafu sice podobného tvaru, ale jsou vertikálně posunuty.



Obr. 6.3: Grafy odběrů pro šifru AES v porovnání s průběhem bez šifrování.



Obr. 6.4: Grafy odběrů pro šifru ChaCha v porovnání s průběhem bez šifrování.



Obr. 6.5: Grafy odběrů pro šifru OTP v porovnání s průběhem bez šifrování.

7 ZÁVĚR

Cílem této práce bylo navrhnout a implementovat řešení zabezpečení komunikace na bezdrátových nízko-energetických technologiích. Konkrétně pro komunikační síť Sigfox. Toho bylo dosaženo postupným získáváním znalostí o LPWAN technologiích obecně a poté přímo o Sigfox síti. Dalším předpokladem bylo seznámit se s kryptografií a detailně zjistit možnosti implementace různých kryptosystému do koncového zařízení. Z řad modernějších šifer jsme popsali a zvážili použití šifer ChaCha, Salsa, AES, Sosemanuk, Panama a OTP.

Po zjištění všech potřebných informací o šifrách jsme se rozhodli implementovat šifry ChaCha, AES a OTP. Šifra AES je bloková šifra s délkou bloku větším než je maximální možná velikost dat, které lze odeslat v jedné Sigfox zprávě. To znamenalo značný problém, který jsme však vyřešili použitím proudového režimu CTR. Ten nám dovoluje zašifrovat libovolnou délku zprávy. Šifra ChaCha je proudová šifra, takže tento problém u ní nenastal. Dále i šifra OTP dovoluje šifrovat zprávy o libovolné délce.

Po výběru šifry následovalo realizování samotných implementací. Abychom mohli šifry implementovat, je potřeba nejdříve vybrat a zapojit vhodný hardware. My se rozhodli použít mikroprocesor ATMEGA328P-PU, který se také používá v zařízeních Arduino UNO. Pro samotné posílání zpráv skrze Sigfox síť je potřeba použít Sigfox modulek. V našem případě byl použit modulek osazen čipem WiSOL SFM10R1. Aby celé zařízení odebíralo co nejméně proudu a vydrželo fungovat na baterii co nejdéle, tak je nezbytné ho uvádět do režimu spánku. Mikroprocesor nabízí několik režimů spánku, ale zdaleka nejúspornější je režim Power Down. Z tohoto režimu lze mikroprocesor probudit pomocí pinů přerušení (interrupt pins). My se rozhodli využít jeden tento pin, skrze který mikroprocesor probouzel tzv. real-time clock obvod s označením MCP7940N I/P. Tento obvod ve své paměti udržuje přesný hodinový čas a datum. Na základě předešlého nastavení probouzel mikroprocesor ze spánku v pravidelných časových intervalech. Jako zdroj přenášených dat jsme si vybrali nízko energetický, digitální teploměr DS18B20, který dokáže měřit teplotu okolí s rozlišením až 12 bitů.

S těmito základními součástmi zařízení odebíralo v režimu spánku proud o velikosti 9 µA. Tuto hodnotu a také závislosti odběru proudu na čase jsme získali z měření, která byla uvedena v závěru práce. Tato měření měla za účel zjistit, která z vybraných šifer je nejúspornější. Po porovnání odběrů všech šifer s odběrem bez šifrování jsme zjistili, že nejúspornější je šifra ChaCha20. Z výsledných grafů se zdá být nejúspornější šifra OTP. Ta však pro svojí kompletní implementaci potřebuje bezpečné paměťové úložiště pro šifrovací klíče, které by odebíralo příliš velké množství nadbytečného proudu (přibližně 10 µA).

LITERATURA

[1] HRSTKA, J. *Síťové technologie LPWAN pro Internet věcí – 1. díl* [online]. 2017, [citováno 20. 5. 2018]. Dostupné z: <http://www.netguru.cz/novinky/3707-sitove-technologie-lpwan-pro-internet-veci-1-dil/>.

[2] HRSTKA, J. *Síťové technologie LPWAN pro Internet věcí – 3. díl* [online]. 2017, [citováno 20. 5. 2018]. Dostupné z: <http://www.netguru.cz/novinky/3772-sitove-technologie-lpwan-pro-internet-veci-3-dil/>.

[3] HRSTKA, J. *Síťové technologie LPWAN pro Internet věcí – 4. díl* [online]. 2017, [citováno 20. 5. 2018]. Dostupné z: <http://www.netguru.cz/novinky/3792-sitove-technologie-lpwan-pro-internet-veci-4-dil/>.

[4] RAKON, L. *Rakon Thinxtra SIGFOX - your questions answered* [online]. [citováno 20. 5. 2018]. Dostupné z: <http://www.rakon.com/component/docman/doc_download/499-rakon-thinxtra-sigfox-your-questions-answered/>.

[5] VOJÁČEK, A. *SIGFOX - princip, struktura, protokol, použití* [online]. [citováno 20. 5. 2018]. Dostupné z: <https://vyvoj.hw.cz/sigfox-princip-struktura-protokol-pouziti.html/>.

[6] WAVIoT *Why WAVIoT uses proprietary narrowband protocol instead of LoRa coding gain* [online]. [citováno 20. 5. 2018]. Dostupné z: <http://waviot.com/blog/technology/why-waviot-uses-proprietary-narrowband-protocol-instead-of-lora-coding-gain.html>.

[7] KRPÁLEK, J. *Inteligentní řízení veřejného osvětlení v koncepci IoT* [online]. [citováno 20. 5. 2018]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/66875/F3-DP-2017-Krpalek-Josef-Inteligentni_rizeni_verejneho_osvetleni_v_koncepci_IoT.pdf?sequence=-1>.

[8] Sigfox S.A. *Sigfox Technical Overview July 2017* [online]. [citováno 20. 5. 2018]. Dostupné z: <https://simplecell.eu/wp-content/uploads/2018/01/Sigfox-technical-overview-July-2017.pdf>.

[9] HUŘŤÁK, P. *Analýza virtuální měny Bitcoin* [online]. Praha, Vysoká škola ekonomická v Praze, 2013, [citováno 20. 5. 2018]. Dostupné z: <http://theses.cz/id/yb0fq4/>.

[10] ŠILHAVÝ, P. *Datová komunikace*, VUT v Brně, první vydání. 2012. ISBN: 978-80-214-4455-3.

[11] ROHLÍK, M. *Využití proudových šifer v současnosti* [online]. 2009, [citováno 20. 2. 2018]. Dostupné z: <http://access.feld.cvut.cz/view.php?cisloclanku=2009080001>.

[12] KLÍMA, V. *Základy moderní kryptologie - Symetrická kryptografie II.* [online]. [citováno 20. 2. 2018]. Dostupné z: <http://crypto-world.info/klima/mffuk/Symetricka_kryptografie_II_2006.pdf>.

[13] TORNEA, O., BORDA, M. E., PILECZKI, V., MALUTAN, R. *DNA Vernam cipher*. E-Health and Bioengineering Conference, 2011.

[14] VALÁŠEK, M. *Symetrické šifrování AES/Rijndael v .NET* [online], 2007, [citováno 20. 2. 2018]. ISSN: 1801-9447. Dostupné z: <http://www.aspnet.cz/Articles/147-symetricke-sifrovani-aes-rijndael-v-net.aspx>.

[15] BERNSTEIN, D. J. *The Salsa20 family of stream ciphers* [online]. New stream cipher designs, 2008, [citováno 20. 2. 2018]. Dostupné z: <https://cr.yp.to/snuffle/salsafamily-20071225.pdf>.

[16] BERNSTEIN, D. J. *ChaCha, a variant of Salsa20* [online], Workshop Record of SASC (Vol. 8, pp. 3-5), [citováno 20. 2. 2018]. Dostupné z: <http://cr.yp.to/chacha.html>.

[17] BERBAIN, C., BILLER, O., CANTEAUT, A., COURTOIS, N., GILBERT, H., GOUBIN, L., GOUGET, A., GRANBOULAN, L., LAURADOUX, C., MINIER, M., PORNIN, T., SIBERT, H.*Sosemanuk, a fast software-oriented stream cipher* [online], New stream cipher designs, 2008, [citováno 20. 2. 2018]. Dostupné z: <http://www.ecrypt.eu.org/stream/p3ciphers/sosemanuk/sosemanuk_p3.pdf>.

[18] DAEMEN, J., CLAPP, C. *Fast hashing and stream Encryption with PANAMA* [online], International Workshop on Fast Software Encryption, 1998, pp. 60-74, [citováno 20. 2. 2018]. Dostupné z: <http://jda.noekeon.org/JDA_CCL_Panama_1998.pdf>.

[19] DAI, W. *Crypto++ 6.0.0 Benchmarks* [online]. [citováno 20. 2. 2018]. Dostupné z: https://www.cryptopp.com/benchmarks.html>.

[20] SCHWARTZ, J. *U.S. Selects a New Encryption Technique* [online], 2000. [citováno 20. 2. 2018]. Dostupné z: <http://www.nytimes.com/2000/10/03/business/technology-us-selects-a-new-encryption-technique.html>.

[21] MORRIS, D. *Recommendation for Block Cipher Modes of Operation: Methods and Techniques, Methods and Techniques* [online], NIST, 2001, [citováno 20. 2. 2018]. Dostupné z: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf>.

[22] Lipmaa, H., Rogaway, P., Wagner, D. *Comments to NIST concerning AES Modes of Operations: CTR-Mode Encryption* [online]. NIST, [citováno 20. 2. 2018]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.297.6301&rep=rep1&type=pdf>.

[23] LANGLEY, A., CHANG, W., MAVROGIANNOPOULOS, N., STROMBERGSON, J., JOSEFSSON, S. *ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)* [online], 2015, [citováno 20. 2. 2018]. Dostupné z: <https://tools.ietf.org/html/draft-ietf-tls-chacha20-poly1305-04>.

[24] MARK, M. *FreeBSD Revision 317015* [online], 2017, [citováno 20. 2. 2018]. Dostupné z: <https://svnweb.freebsd.org/base?view=revision&revision=r317015>.

[25] GUENTHER, P. *ChaCha based random number generator for OpenBSD* [online], 2015, [citováno 20. 2. 2018]. Dostupné z: <http://bxr.su/OpenBSD/lib/libc/crypt/arc4random.c>.

[26] CAMPELL, T. *Legacy arc4random(3) API from OpenBSD reimplemented using the ChaCha20 PRF, with per-thread state* [online]. 2016, [citováno 20. 2. 2018]. Dostupné z: <http://bxr.su/NetBSD/lib/libc/gen/arc4random.c>.

[27] AUMASSON, J. P., FISCHER, S., KHAZAEI, S., MEIER, W., RECHBERGER, C.*New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba* [online]. 2007, [citováno 20. 2. 2018]. Dostupné z: <https://eprint.iacr.org/2007/472.pdf>.

[28] RIVEST, R. *Computer and Network Security Lecture 2* [online]. 1997, [citováno 20. 2. 2018]. Dostupné z: <http://web.mit.edu/6.857/OldStuff/Fall97/lectures/lecture2.pdf>.

[29] Rosulek, M. *The Joy of Cryptography* [online]. School of Electrical Engineering and Computer Science Oregon State University, 2018, [citováno 20. 2. 2018]. Dostupné z: <http://web.engr.oregonstate.edu/~rosulekm/crypto/chap1.pdf>.

[30] Atmel Corporation. *8-bit AVR Microcontrollers ATmega328/P DATASHEET COMPLETE* [online]. 2016, [citováno 20. 2. 2018]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/

Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf>.

[31] Anonymous Author. *On Track of Sigfox Confidentiality with End-to-End Encryption*. ARES-IoT-SECFOR'2018, Hamburg, Germany, 2018, [citováno 20. 2. 2018]. DOI: 10.475/123_4.

[32] Atmel Corporation. *AT27LV040A 4Mb (512K x 8) Low Voltage, One-time Programmable, Read-only Memory* [online], 2011, [citováno 20. 2. 2018]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/doc0557.pdf>.

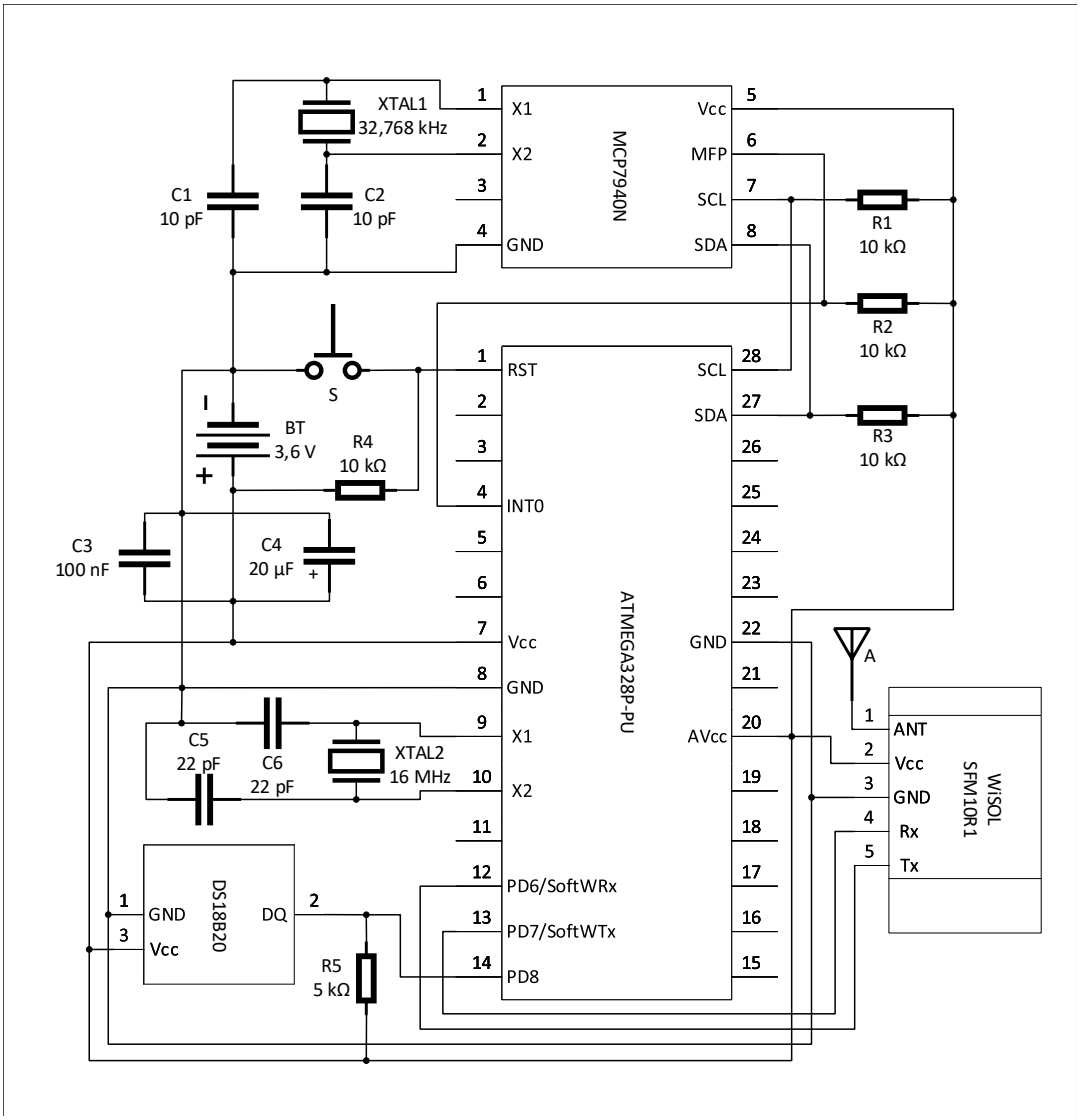
[33] WEATHERLEY, R. *Arduino Cryptography Library* [online], 2018, [citováno 20. 2. 2018]. Dostupné z: <https://rweather.github.io/arduinolibs/crypto.html>.

[34] NIR, Y., LANGLEY, A. *ChaCha20 and Poly1305 for IETF Protocols* [online]. 2015, ISSN: 2070-1721, [citováno 20. 2. 2018]. Dostupné z: <https://tools.ietf.org/html/rfc7539>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

LPWAN	Low-power Wide Area Network
UNB	Ultra Narrow Band
SS	Spread Spectrum
LoRaWAN	Long Range Wide Area Network
RPMA	Random Phase Multiple Access
ISM	Industrial, Scientific and Medical
DBPSK	Differential Binary Phase-Shift Keying
TCP/IP	Transmission Control Protokol/Internet Protokol
IT	Information Technology
MAC	Media Access Control
SSL	Secure Sockets Layer
VPN	Virtual Private Network
K_E	Šifrovací klíč
K_D	Dešifrovací klíč
IV	Initialization Vector
S/MIME	Secure/Multipurpose Internet Mail Extensions
GSM	Global System for Mobile
ECB	Electronic Codebook
CBC	Cipher Block Chaining
CTR	Counter
RAM	Random Access Memory
OTP	One-Time Pad
XOR	Exclusive Or
HTTPS	Hyper Text Transfer Protocol Secure
WEP	Wired Equivalent Privacy
WPA	Wireless Protected Access
PWM	Pulse Width Modulation
GND	Ground
RTC	Real Time Clock
IRTF	Internet Research Task Force

A SCHÉMA ZAPOJENÍ



Obr. A.1: Příklad nasazení v praxi - schéma zapojení.

B OBSAH PŘILOŽENÉHO DVD

Přiložené DVD obsahuje elektronickou verzi této bakalářské práce, zdrojové programy šifer (režim odesílání jednou za hodinu) a knihovnu, z které byly použity programy šifer AES CTR a ChaCha. Zdrojové programy šifer jsou spustitelné v programu Arduino IDE a společně s přiloženou knihovnou byly testovány v programu Arduino IDE verze 1.8.5. Ostatní knihovny, které byly použity v programech jsou součástí Arduino IDE nebo je lze stáhnout přímo z Arduino IDE (Projekt –> Přídat knihovnu –> Spravovat knihovny...).

```
/ ..... kořenový adresář přiloženého DVD
├── Bakalarska_prace_xblaze32.pdf ..... elektronická verze této práce
├── Bakalarska_prace_AES-CTR ..... zdrojový kód programu šifry AES CTR
│   └── Bakalarska_prace_AES-CTR.ino
├── Bakalarska_prace_ChaCha ..... zdrojový kód programu šifry ChaCha
│   └── Bakalarska_prace_ChaCha.ino
├── Bakalarska_prace_OTP ..... zdrojový kód programu šifry OTP
│   └── Bakalarska_prace_OTP.ino
└── Arduino Cryptography Library ..... Arduino IDE knihovna
    └── arduinolibs-master.zip ..... autor Rhys Weatherley [33]
```